Cody Coyote's

Mission Building Handbook

(Revised April, 2011)

A Comprehensive Guide to Building Missions and Campaigns in Microsoft's®
Combat Flight Simulator 2

Table of Contents

Introduction.....1

Part One – Basic Mission Building

Chapter One – Getting Started.....3
Introduction to mission building
Creating a desktop shortcut

Chapter Two – Beginner's Luck.....5
Designing a simple mission using waypoints, spawning and warping

Chapter Three – Raid on Tarawa.....19
Placing infrastructure
Use of Event Areas
More complex triggers and events

Chapter Four – A "Wake" Up Call.....40
Multiple waves of attackers
Chart filters
Multiple mission endings
Ditching

Part Two – Enhanced Mission Building I

Chapter Five – Playing the Percentages.....63
Using random chance and percentages
Hiding objects from the player

Chapter Six – Avoiding The Never Ending Mission.....75
Alternate route waypoints
Landing as a mission goal

Chapter Seven – Customized Debriefings.....84
Multiple endings – multiple debriefings
Limitations on briefings and debriefings

Chapter Eight – Missions Without Waypoints.....87
Designing missions without using waypoints
Limiting waypoint use

Chapter Nine – Moving Objects.....95
Ships – convoys, sailing north, running over land
Stopping landing craft
Ground Units (Vehicles)

Chapter Ten – Infrastructure.....103
Use of infrastructure – placing, grouping, copying, orientation
Layouts

Chapter Eleven – Front Lines.....115
Placing and using front lines
Effect on triggers, events and mission goals

Chapter Twelve – Sounds.....118
CFS2 sounds – locations and functions.
Overview of using sounds in Mission Builder
Sounds in the ETO

Part Three – Enhanced Mission Building II

Chapter Thirteen – Hand-Editing Basics.....122
Basics of hand-editing
The MIS file
The DYN file

Chapter Fourteen – Weather & Wingmen.....131
Custom weather including snow
Flying as a Wingman
Making your wingmen all different planes
Flights of more than eight aircraft

Chapter Fifteen – How to Take Off from any Point on the Planet.....146
Creating a takeoff point without the use of the airbases.dat file

Chapter Sixteen – Error Messages.....167
Avoiding errors in the first place
Common error messages

Part Four – Campaign Building and Beyond

Chapter Seventeen – Writing an Effective Campaign.....178
The Linear Campaign
The Dynamic (Branched) Campaign
Movies and Sounds
Troubleshooting

Chapter Eighteen – Other Related Files.....196
airbases.dat
callsign.dat
country.cfg
descrip.dat
DP files
INFO images
mdlnames.dat
opareas.dat
outcome.dat
scuttlebutt.dat
Readme

Chapter Nineteen – Mission Accomplished.....207
Improving AI performance
Acknowledgements
Legal

Appendix A.....210
Files included with this Handbook

Appendix B.....211
Transcripts of Radio Transmissions used in Mission Builder

Appendix C.....225
Ten Downloads I Wouldn't be Without

Appendix D.....227
Websites Worth Visiting

Introduction

When Microsoft's® Combat Flight Simulator 2 (CFS2) first came out I was just a pup. Who would have thought that ten plus years later it would still be going strong?

In dog years I'm almost 77, a senior citizen. In computer years CFS2 is, well, ancient! 10 years is a lifetime for anything having to do with computers in this day and age. My digital camera has more memory then the first PC I installed CFS2 on. By all rights CFS2 should have fallen by the wayside long ago, yet it has developed a faithful following of dedicated flight simmers who continue to develop addons and push the envelope. Whether you're an amateur historian looking to recreate the famous air battles of World War II or just a dedicated gamer who likes to fly combat missions, CFS2 has something to offer.

Shortly after I purchased CFS2 I became fascinated with the Mission Builder but immediately noticed that there was no tutorial to teach the beginner all of the ins and outs of mission building. In 2002 I released Cody Coyote's Mission Building 101*, my first in a series of tutorials aimed at helping the novice master mission building. Over the years that tutorial was very well received and I thank all of you who downloaded a copy and found it useful.*

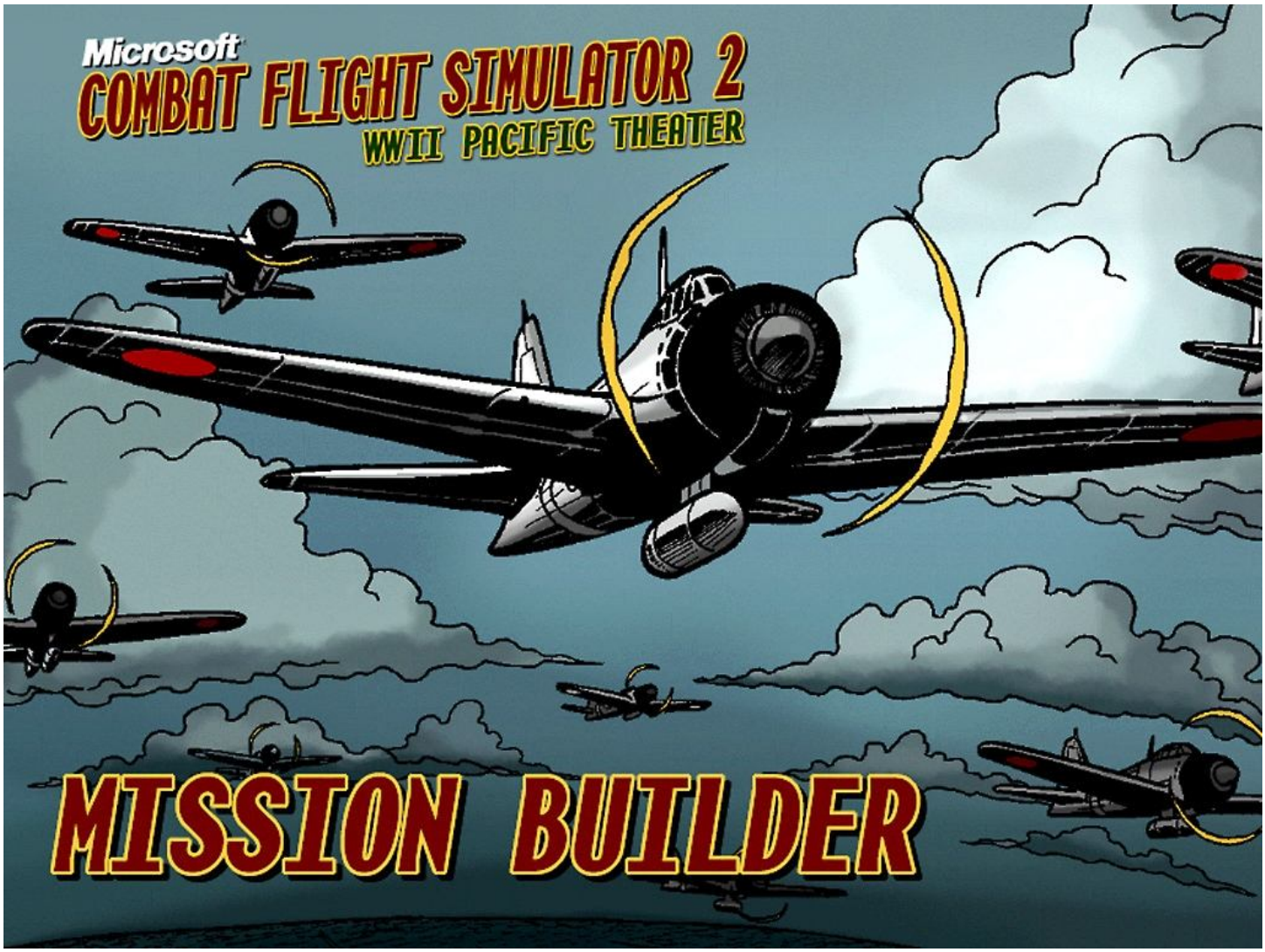
In 2007 I first published this Handbook. It was more than just a collection of my various mission building tutorials under a single cover. It contained much additional information never published before and was a reflection of the lessons I'd learned in the thousands of missions I'd written in the previous seven years.

Today, over ten years into it's release, CFS2 is still going strong and new players are discovering it for the first time or rediscovering it after have been away. We are even seeing new websites dedicated to CFS2.

As a result of this continuing interest, I decided to update my Handbook and release it again, hopefully to some players just joining our ranks. Much of the original Handbook remains unchanged; the lessons and techniques taught form a solid base on which to grow your skills. I have added a new section in Chapter 14 that covers how to increase the size of your flights beyond eight planes and updated the appendixes covering downloads and websites. You'll find one or two other new things or minor changes along the way as well.

Whether you're new to CFS2 or an old hand I think you'll find this Handbook valuable. Like all of my works, it takes you step by step through each process so that you can easily accomplish the goals of each chapter. I've tried to put something in here for everyone, regardless of your experience level. Enjoy! - April, 2011

Cody Coyote
cody_coyote2000@hotmail.com



Part One

Basic Mission Building

Part One takes you step by step through the design and construction process of three progressively more complex missions. By the end of this section you will be familiar with basic mission design, warping, mission goals, and the use of triggers and events.

Chapter One – Getting Started

Purpose of this Handbook
I should tell you right up front what this Handbook is and is not.

It **is**:

- a comprehensive guide to mission and campaign building
- a series of step by step instructions designed to take you from basic through more advanced mission building and campaign design
- a confidence builder for simmers of any experience level who want to get enjoyment, not frustration out of designing their own missions
- a tool for more experienced users on how to hand-edit their missions
- a guide to some techniques, tips, and tricks which will improve your mission building experience regardless of your experience level

It is **not**:

- the final word in mission design or technique

Some Background and Ground Rules
Free Flight and Quick Combat offer you the opportunity to fly your favorite planes anywhere in the world or against other planes but to get the real feel of CFS2 you need to fly missions. Fortunately the game comes with a variety of training and combat missions already installed. In addition there are two campaigns documenting the air war in the Pacific, one from the U.S. side and one from the Japanese perspective. All are good, but when you fly them over and over the challenge goes away and they become old hat. The solution is to add new and challenging missions to your game. Fortunately there is no shortage of simmers out there willing to write new missions and offer them for free over the Internet. That's fine as far as it goes, but if you're like me you'll want to take a crack at designing your own missions using your own ideas and your favorite planes. Microsoft anticipated that desire and included a sophisticated mission builder with CFS2.

When I first opened Mission Builder, I quickly read everything I could find looking for the tutorial. I was disappointed to discover that there wasn't one. There is a comprehensive Help file that, while informative, is not a tutorial and makes assumptions about your knowledge of mission structure, which may or may not be true.

As I struggled to teach myself how to build missions I also looked to the Internet and the myriad of CFS2 websites out there. While there were some tutorials available, there were not a lot of them and I'm afraid I found all of them lacking. The ones I saw either addressed 'advanced' aspects of mission building or, like the help files, assumed you knew things about mission design that you may not. What I was looking for and couldn't find was a simple guide for the beginner. A handbook for the simmer who wishes to get started building their own missions and doesn't know where to begin. I decided if I could teach it to myself, I could probably teach it to you, so I opted to write my own 'How To' guide.

This Handbook will lead you step by step through a series of increasingly more complex missions. Each mission is intended to build on the previous one and introduce different techniques and features of the Mission Builder. This is followed by chapters covering how to combine your missions into a campaign, and other techniques and tricks designed to help you get the most out of your mission

building experience. When we're finished, I hope you will feel comfortable with the Mission Builder and inspired to learn more. I merely want to open the door for you to having some additional fun with what is already a terrific game.

Before we begin you should be familiar with the basic computer commands such as Open, Copy, Paste, etc. Also, from time to time I will refer to opening a mission file or other document. I refer to Notepad in the text but you are welcome to use any text editor of your choosing. Finally, any reference to file locations assumes your copy of CFS2 is installed in the default location of: *C:\Program Files\Microsoft Games\Combat Flight Simulator 2*.

Getting Started

Before you can write your first mission you need to activate the Mission Builder. If you did a full install of CFS2 you'll find the icon for Mission Builder under START→All Programs→Microsoft Games→Combat Flight Simulator 2→Mission Builder. Simply click the icon to launch Mission Builder. If you don't see the Mission Builder icon please refer to the sidebar below for instructions on creating a desktop shortcut.

Creating a Desktop Shortcut for Mission Builder

If you can't find your Mission Builder shortcut you can easily create one and place it on your desktop.

- Go to **start →My Computer** and navigate to the default CFS2 location: **C:\Program Files\Microsoft Games\Combat Flight Simulator 2**
- Right click on the **cfs2** icon (Looks like a pilot's head) and click again on **Send To → Desktop (create shortcut)**
- Go to your Desktop and look for the icon named **Shortcut to cfs2**
- Right click on the icon and then click on **properties**
- In the **Target** window add a space after the last character then type in: **/BUILDER:**
- When you are finished the new target should look like this: **"C:\Program Files\Microsoft Games\Combat Flight Simulator 2\cfs2.exe" /BUILDER:**
- Be sure to have the space between the end of **exe"** and the slash in front of **BUILDER**
- Click **Apply** then **OK** and close out the window
- Rename the new icon to **CFS2 Mission Builder** or whatever you'd like.
- Now all you have to do is double-click your new icon and Mission Builder will start

These instructions assume CFS2 is located in the default location, if you installed it somewhere else you will need to change the paths accordingly.

CFS2, Mission Builder and Windows 7

Although released many years before Windows 7, CFS2 and the Mission Builder run just fine under this latest version of Microsoft's operating system. I'm currently running Windows 7 Professional 64 bit without a hitch. If you are having problems, I'd first check to make sure you are running with Administrator privileges. If the problems persist, I'd look elsewhere. I don't believe Windows 7 is the culprit.

Chapter TWO – Beginner's Luck

Preflight

The very first thing you should do, even before starting Mission Builder, is to decide on what you want your mission to be. Come up with an idea. You don't need to have every plane and every move plotted out in your head but you should have a clear, general idea of what you want to do. A mental picture of what you want your mission to look like will help you build a mission that meets that vision. Your initial idea can be pretty simple:

- Lead a flight of Wildcats as they protect US bombers attacking a Japanese convoy.
- Take off and engage a flight of enemy fighters attacking your base.
- Fly a solo attack against U.S. ground targets

Get something in your head so you have a starting point. On complex missions you may want to jot down a few notes or create an outline of what actions you want to include, the aircraft types, locations, etc. For historical missions, you will need to do the appropriate research. Do what works best for you as long as you have some idea of what you're trying to build.

Now that you have your idea, power up Mission Builder. As mentioned above, you should find it under START→All Programs→Microsoft Games→Combat Flight Simulator 2→Mission Builder. If you plan to use Mission Builder a lot you may want to create a desktop shortcut or a Quick Launch button for it. I found it easiest to add a shortcut to the top of the Start menu. Wherever you put it, find it. Start it up. Wait for it to load up then proceed.

This is the main Mission Builder screen and the point from which all mission building starts. The screen is divided into two parts. The left third of the screen displays information on the type of equipment, aircraft, ship, vehicle, and infrastructure that you have highlighted, while the right two thirds of the screen is a map of the area in which your mission is taking place. The map is orientated so that North is always at the top.

Buttons along the top and bottom of the screen are used for performing various commands and functions used in mission building. Descriptions of these buttons and their use can be found in the Help files. You should become familiar with the Help files and their use as you need to know where to go when a question arises. An understanding of the Help files and their contents will enhance your mission building experience.

The default starting point for any new mission places you (the player) in an A6M2 Zero flying north, about 75 nautical miles East of Munda, right about in the middle of The Slot. I'll guess that 99% of the time you will want to start your mission either somewhere else or in a different airplane. So let's start writing our mission.

For our first mission we are going to fly a Wildcat out of Henderson, intercept a flight of Zeros, shoot them down (hopefully) and return to base. Simple.

Begin by clicking the New Mission button in the upper left-hand corner. This will bring up this window which allows you to start at the current location, an exact location specified by longitude and latitude, an area within the Pacific or a specific airbase. Click on **Airbase**, scroll down to **Henderson**, highlight it, and click on **OK**.

The map now places you at Henderson, still flying North in a Zero however. First let's change planes.

On the left hand side of the screen under 'Player Aircraft' you will see a column with the following fields lit up, **Aircraft**, **Nationality**, **Mission**, **Unit ID**, **Payload**, **# of aircraft**, and down near the bottom **Pre damage (%)**.

Click the down arrow next to the Aircraft window and scroll down until you find the **F4F-4 Wildcat** and select it. Follow the same pattern to select **USA** as your nationality. Under 'Mission' leave it as **Fighter Sweep**. The eight different types of missions are listed and explained on page 139 of your CFS2 Pilot's Manual. Refer to that book for more information. If you'd like to give yourself a name in the mission, scroll down under 'Unit ID' and select one you like. Since you will be flying alone it isn't really necessary to identify yourself. Leave the payload as **Guns Only** and the number of aircraft as **1**. You're on your own this first time! Leave 'Pre-damage' set to **0**. You can start any plane out with existing damage by setting a value in this box. You might use this if you were to design a mission where you were escorting a damaged bomber back to base, as we will do in Chapter 3, for example.

Since you are flying alone the 'Formation' box is not lit up. As the player, your own ability determines your skill and aggressiveness levels so you can never change these. At the same time, you as the player must always survive to complete the mission so you don't have a choice there either. OK, now you're flying a Wildcat, so let's fly it someplace.

Click on the middle, Route, tab. This will change the window to look like the illustration. Using this panel and the map we will build the waypoints for the mission. Waypoints mark the route that planes, ships and vehicles must follow in the game. Generally, you must pass through each designated waypoint to complete a mission.

Before we proceed, notice the section of this panel marked 'Altitude'. 'BALT' is highlighted and the altitude shows as 3674 feet. Click on 'AGL' and notice that the altitude drops to 164 feet. BALT is the altitude you are above sea level. AGL is the altitude you are above the ground directly below you. Looking on the map you see that your plane is starting out located over land. Since you are 3674 feet above sea level but only 164 feet above the ground, you are obviously flying over elevated terrain. You need to have a basic understanding of the difference between BALT and AGL. If you only focus on the BALT you could end up starting your mission by flying right into the side of a mountain.

Let's begin our mission by taking off from Henderson. Click the down arrow next to 'Action' and select **Take Off**. Notice that when you did this your plane jumped to the airfield marker, another window opened showing your take off point as Henderson and the altitude options were disabled.

Turning to the map, click the MINUS (-) sign at the bottom of the map one time. This will zoom our view out a little and give us more room to fly around in.

Notice that there is a little dot mid-way between your plane and your only waypoint (#2). Place your cursor on this dot, left click and drag it to the right, about midway between the two large islands and close to the edge of the map. Notice that when you clicked on the dot it became waypoint #2 and what was waypoint #2 is now #3. Now click on the dot between waypoints 2 & 3 and drag it due North a little toward the large island. Notice that the old waypoint #3 is now #4.

Left click on waypoint #2 to highlight it and go back to the left-hand side of the screen. It should show waypoint 2/4, the action as turn and the altitude as 717 feet. Using the up arrow, change the altitude for waypoint #2 to 3500 feet. Leave the airspeed alone. Then **click on the 'Warp To' box**. This is very important. You can't warp to a waypoint unless you tell the game that it's OK to do so.

Getting Your Bearings

Do you know where you're at?

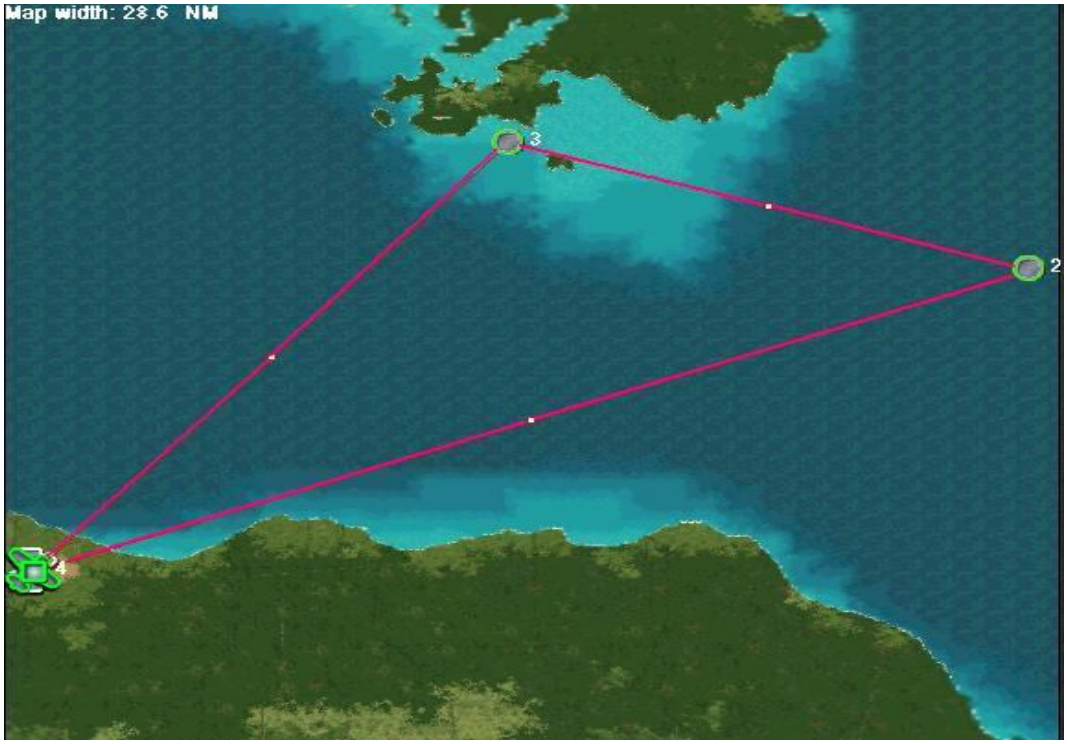
Henderson is located on Guadalcanal Island in the Solomon's. The large island just to the northeast is Florida Island and the small round island to the northwest is Savo Island. The water you'll be flying over is known as "Iron Bottom Sound" because of all of the ships sunk there during the battle for Guadalcanal.

Move up and click the right hand arrow next to where it says Waypoint 2/4. This will move you to waypoint #3 (alternatively you could left click waypoint #3 on the map). The altitude again is 717 feet. Change that to 1000 feet. Click on the 'Warp To' box then move to waypoint 4/4.

Under 'Action' select 'Land at Base'. Notice that when you did that waypoint #4 moved down to Henderson and changed into a green diamond. Refer to the illustrations for guidance.

Plotting waypoint #2

Plotting waypoint #3



All four waypoints plotted

Congratulations! You just plotted the flight plan for your first mission. Now, let's add some enemy planes, a few triggers, an event or two and get flying.

Click on the (**+**) button along the bottom toolbar below the map. This will change your cursor to a magnifying glass so that you can zoom in. Place the cursor over waypoint #2 and click twice. Your map width will be about 20 miles. Now, click on the **New Aircraft** button on the tool bar above the map. This will change your cursor into an airplane. Place the plane on top of your flight path about an inch to the left of your waypoint #2 and left click. Then drag the cursor to a point a slightly below and well to the right of waypoint #2. Left click again. Finally, right click to return your cursor to an arrow. Good, you have just added another plane and given it two waypoints – starting and ending.

Now we need to turn this plane into a target. Click on the new plane's starting point. Move to the information panel on the left side of the screen and make the following changes. Select the **A6M5 ZERO** as the type of aircraft. Set the nationality as **Japan**. Leave the mission as **Fighter Sweep**. Choose **Honda** as the unit ID and the payload as **Guns Only**. Hit the up arrow next to number of aircraft and change that to **2**. You can select any formation you wish but for simplicity's sake leave it as the default of **Fingertip**. Leave the skill level as **Rookie** and set the **Aggressiveness** to **Average**. Finally select **Must be Destroyed** as the Goal. By selecting 'Must be Destroyed' you have made that a mission goal. For you to successfully complete this mission, both enemy planes must be destroyed. Choosing 'Must Survive' means that the object must survive to successfully complete the mission. Choosing 'None' as a goal means it doesn't matter what happens to the object it won't count as a goal. Ignore the remainder of the fields.

As you gain proficiency, experiment with the different levels and see how it impacts the mission. Since this is your first mission and I want you to complete it successfully we're using a fairly easy setting of 'Rookie' and 'Average'.

Click on the **Route** tab. Make sure it is showing waypoint #1. If not, click the left or right arrows to get there. The action will show as **Turn**. Leave that as is. Drop down to the altitude and change it to 3500 feet. We want the enemy planes and our plane to be at the same altitude so that we can see them and engage them easier. Go to waypoint #2 and change that altitude to 3500 feet also. *Although, with luck, those Zeros will never get to waypoint #2*!

Now we need to get to your waypoint #3. There are three ways we can do that.

- Using the minus (-) button below the map, back out until waypoint #3 becomes visible. Then zoom in on it like you did for waypoint #2.
- Click on the 'Pan' button (the one that looks like a hand) then holding the left mouse button down, drag the map down and over to the right until waypoint #3 comes into view.
- Click on waypoint #2; go to the information panel on the left, click on the route tab then using the right arrow go to waypoint #3.

Choose one then meet me at waypoint #3.

We are now going to add another target to shoot up on the way back to the base. Click on the **New Ship** button along the top. Your cursor will turn into a ship silhouette. Place the first waypoint for the ship along your flight path, just in front of your waypoint #3 then drag the cursor, placing the second waypoint along your flight line but farther out. Right click to finish. This will give the effect of approaching the ship from the rear.

Click on the ship's starting point then go over to the left side of the screen and we'll set up this target. Scroll down, under 'Ship Type' select **HIJMS Tidori class Torpedo Boat**. You can choose any vessel you want but it might be tough to sink an aircraft carrier with just guns so let's keep it easy for this first mission. Much like with the Zeros, select the other fields, 'Nationality' – **Japan**, 'Unit ID' – **PB75**, 'Skill' – **Novice**, and 'Goal' – **None**. You now have three goals you must achieve to successfully complete this mission:

1. You must survive – this is always a goal.
2. Shoot down the first Zero
3. Shoot down the second Zero

Note that sinking the patrol boat is not a goal since we selected 'None' instead of 'Must be destroyed'. OK, let's link everything together with some triggers and events, write up some accompanying text and go flying!

The complete mission shown plotted on the map.

A Few Words about Triggers and Events

Triggers and events are what give life to your missions. They can be simple or complex and can cause a mission to go different directions each time it is flown. Mastering triggers and events will allow you to design more exciting and challenging missions. The first three sample missions in this Handbook use simple triggers and events. To learn more, refer to some of the missions that came with CFS2. Open them up in Mission Builder and look at the different triggers and events. See how they interact and what effect they have on the game. You can also refer to Chapter Seven – Missions Without Waypoints. Finally, experiment on your own as you build experience and confidence.

An **event** is something that happens. A **trigger** is the thing that causes the event to happen. An event cannot happen until it has been triggered. We have triggers and events in our daily lives only we don't think of them in those terms. Trigger: the alarm goes off, Event: you wake up and get out of bed; Trigger: a police officer pulls you over, Event: your blood pressure rises; Trigger: you see a beautiful woman walking down the street, Event: your blood pressure rises. Notice that in the last two examples the same type of event can have different triggers.

In CFS2 an event might be enabling warp, the trigger could be arriving at a certain waypoint. Just like in the example above, the same type of event could have two different triggers. The event may be ending the mission. The triggers could be destroying all of the targets or leaving the combat area. Which one happens first would determine which trigger is used.

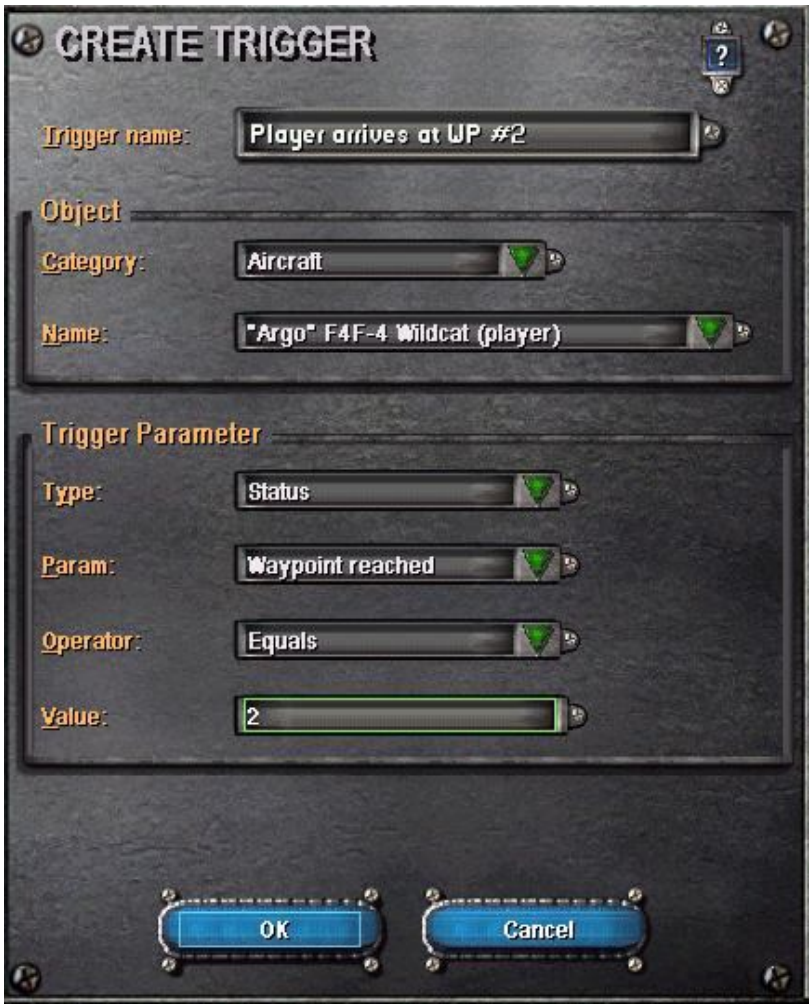
Adding Triggers and Events to our Mission

It helps to first decide what events you want to occur in your mission and then build the triggers to support those events. Our mission is a very simple one. We want these events to occur:

1. warp to waypoint #2
2. spawn the enemy planes
3. warp to waypoint #3
4. spawn the enemy ship
5. warp to waypoint #4
6. end the mission

It isn't necessary to warp to a waypoint, you can fly there if you want to in real time, but it is nice to build the warp option into the mission. By the same token, you don't have to spawn the enemy targets. However, if you don't spawn them they will appear when the mission starts and they may have flown or sailed away from where you want to meet them by the time you get there. By spawning the targets we insure that they will be where we want them to be when we get there.

We have six events so we need six triggers right? Not quite. Mission Builder is set up so that the first warp is automatic, we don't have to create a trigger for that one. Also, when you get to the last waypoint you can end the mission by pressing "X". These automatic warps and mission endings will occur when there are no enemy targets in the area or you have not disabled warp. On missions where you start by taking off, you must first achieve a minimum altitude and air speed before the warp will be enabled. So, while we have six events we only need to create 4 triggers to get the mission to work. Let's take them in order.



The first event we want to trigger is spawning the Zeros. Start by clicking the **Triggers** button on the left side of the screen along the top then click **New**. Give the trigger a name. "Spawning Zeros" is NOT the right name. That is a good name for the event, not the trigger. We'll call this trigger **Player arrives at WP #2**. Under 'Object' the default category is **Aircraft** and the player aircraft is already selected. Under 'Trigger Parameter' choose 'Type' – **Status**, 'Param' – **Waypoint Reached**, 'Operator' – **Equals**, 'Value' – **+2**. Look it over and you'll see the trigger we've created is our arriving at waypoint #2. Click **OK** then **Close**.

Next, click on the ‘Events’ button and select ‘New’. Name the event **Spawn enemy Planes**. You will see a window with a list of your triggers for this mission. Right now there is only the one we just created. Highlight it then click on the right arrows to insert it into the right hand column. Drop down on the left-hand side to where it says ‘Actions Type’.

Click the down arrow and select **Miscellaneous**. Click on **Disable warp** then click the double right hand arrows (**>>**). Go back to the drop-down menu under ‘Actions Type’ and select **Spawning**. This gives you two options, ‘Re-spawn unit’ and ‘Spawn unit’. Highlight **Spawn unit**. In the window to the right ‘Honda’ will show as the default. Click the double right hand arrows (**>>**) then click **OK**.

What you’ve just done is created a trigger and an event that combined will disable warp when you arrive at waypoint #2 and cause the two Zeros to appear. Easy wasn’t it?

Now let’s do the next three triggers and events.

Open the triggers and click on 'New'. Title this trigger **Enemy planes destroyed**. Make the following selections:

'Category' - 'Aircraft'
'Name' - 'Honda'
'Type' - 'Status'
'Param' - 'Object Health'
'Operator' - 'Equals'
'Value' - '0'.

Save it and open up a new event. Name this event **Enable warp 1**. In this event we are going to enable warp after the Zeros are destroyed. To do that correctly we need to use two triggers. First, in the 'Triggers Available' window select the event **Spawn Enemy Planes** and add it to the right. Next select the trigger **Enemy planes destroyed** and add it to the right. Next select **Action delay** and set the delay for **5** seconds. Then select **Miscellaneous** under 'Action type' and select **enable warp**. Notice that we are using the first event we created as a trigger in our second event. See the sidebar below for a detailed explanation as to why. Save this event and go back to Triggers.

Logic and Warping

It is critical in this mission to include the event 'Spawn Enemy Planes' as one of the triggers to enabling warp. Why? Follow along and it will all become clear…

Remember we said earlier that an event can also be used as a trigger. What we want to do here is enable warping after you shoot down the two Zeros. Our main trigger for that event is when Honda's Object Health equals zero. To you and me, we know that means that they were shot down but that isn't exactly how the game views it. To the game, **Status – Object Health – Equals – 0** means it can't find that object (Honda) in the game. Therefore, if Honda is not in the game its Object Health must be zero. If not, the game would see it. If we failed to include the event 'Spawn Enemy Planes' as one of our triggers and only used 'Enemy Planes Destroyed' here's what would happen:

When the mission starts the game looks around and can't find Honda. It can't find it because Honda hasn't been spawned yet, but that doesn't matter to the game. All the game knows is that Honda isn't there and immediately the trigger for the event "Enable Warp 1" is found to be true and warp is enabled. You don't know this because you are sitting on the ground waiting to take off. Why is it enabled? Because when the game starts Honda hasn't been spawned yet, this means that the game can't find Honda. And if the game can't find Honda then the Object Health for Honda must be "0"; meaning the trigger "Enemy planes destroyed" must be true and therefore warp is enabled.

Next you take off and get a warp message. This is the automatic warp message you always get after taking off so you press 'X' and jump to waypoint #2. When you arrive at waypoint #2 Honda spawns, you shoot them down, and now you're stuck there because the event that enables warping to waypoint #3 has already occurred when the mission started and an event will only occur once unless you check the "Repeat action" box. Unless you know the logic the game uses you're sitting there in the cockpit wondering why you can't warp.

Logic and Warping (con't)

The solution is a simple one; always include the spawning event as a trigger for the enabling warp event. If you get stuck at a waypoint and can't warp, always go back and work through the logic of your triggers and events. Remember these three simple rules:

1. **An event will only occur once unless you check the "Repeat action" box.**
2. **An Object's Health equals '0' until it is spawned.**
3. **When an event is based on the destruction of a spawned object, always include spawning the object as one of the triggers.**

Although an event occurs when all of the conditions are met (all of the triggers are "true"), I find it best to enter those triggers in chronological order. In other words, while it should work if you entered **Enemy planes destroyed** first then **Spawn Enemy Planes** in the "In use" window of the event, I believe it is better to enter them in proper sequence with the spawning before the destroying.

The next trigger will be similar to the first. Name this trigger **Player arrives at WP #3**. Set it up like you did the first one with the player arriving at waypoint **+3**. Your selections will be:

'Category' - 'Aircraft'
'Name' - 'Argo' (the player's aircraft)
'Type' - 'Status'
'Param' - 'Waypoint Reached'
'Operator' - 'Equals'
'Value' - '+3'.

Click 'OK', close this trigger and open a new event. Title this event **Spawn enemy ship**. Select the new trigger you just made then, just like you did with your first event, select **Miscellaneous**, **Disable warp**, **Spawning**, **Spawn unit** and choose the torpedo boat (PB-75). It will be found below 'Honda' in the drop-down menu.

Now we'll create one last trigger. Title this last trigger **Enemy ship destroyed**. Select the following:

'Category' - 'Moving Object'
'Name' - 'PB-75' (or whatever ID you chose for the torpedo boat)
'Type' - 'Status'
'Param' - 'Object Health'
'Operator' - 'Equals'
'Value' - '0'

Close it and open a new event. Name this event **Enable warp 2**. Just like with the event 'Enable Warp 1', we will use the event spawning the torpedo boat as one of our triggers. Choose **Spawn Enemy Ship** and **Enemy ship destroyed** as your two triggers. Set an action delay of **5** seconds, then choose **Enable warp** under **Miscellaneous**. Next, select **Debrief text** from the dropdown menu under 'Action Type'. Highlight it and click the double arrows. Type in this text:

Congratulations! You have successfully completed the first mission you wrote using Mission Builder.

Click **OK** and save this event. You should have created four triggers and four events as follows:

Triggers	Events
Player arrives at WP #2	Spawn enemy planes
Enemy planes destroyed	Enable warp 1
Player arrives at WP #3	Spawn enemy ship
Enemy ship destroyed	Enable warp 2

This completes the meat of the mission. The only thing we have left to do is flesh out the introductory texts. Before we do that however, let's see if this thing works. Click on the **Fly Now** button. A popup window will appear asking you to save your mission. Title this mission **Beginner's Luck**, that is the name that will appear on the missions list. The file name is how it will be stored on the computer. Call it anything you like, perhaps your initials and the number of the mission. I might use CC_01 for example. Click **Save** and Mission Builder will save the mission and switch over to CFS2. When the mission ends you will automatically return to Mission Builder. Go flying, then meet me back here.

Well, how was it? Did you successfully complete the mission? I forgot to tell you that those Japanese torpedo boat gunners are pretty good shots! I hope you made it home safely. What about all that radio chatter and the messages on your screen? Where did they come from? A variety of canned messages and radio sounds are hard wired into CFS2. The game grabs appropriate ones and uses them to add realism to your flight. This also keeps you from having to reenter some of those routine messages and sounds over and over again into your missions. We'll cover adding custom radio chatter later in this Handbook.

There are two things I hope you noticed. First, when you arrived at Waypoint #2 you saw an on-screen message that said "New mission goal!" This message was caused by the Zeros being spawned at that moment. You'll recall that we set the goal for the Zeros as "Must be destroyed". That wasn't a goal in the game until they were spawned. As soon as they were spawned CFS2 notifies you that you now have an additional goal in your mission. If you want, you could pause the game when this message appears by pressing the ALT key; then click on World, Advanced Info. On the Advanced Information screen click the tab marked "Goals" and you will see a list of the goals you must meet to complete the mission. Surviving is always listed as a goal. Second, notice that although you wrote a debriefing text it did not appear on the screen at the end of the mission. Missions flown within Mission Builder do not display the debrief screen. Don't worry; you'll see it when you fly the mission in the game.

If you followed these steps exactly you were able to take off from Henderson and warp to waypoint #2. After that you were on your own. The Zeros appeared and if you shot down both of them, warp was enabled so you could go to waypoint #3. Once there you could dive down and strafe the torpedo boat. If you sunk her, warp was again enabled to take you home where the mission ended.

Now is the perfect time to tweak the mission. If the Zeros were too close or too far away when they appeared then simply move their waypoint #1 on the map to fine-tune them. Do the same thing for the ship. If either the planes or the ship failed to appear, go back to the triggers and events and compare what you typed in with these instructions. Something doesn't match. Once you have it just the way you want it we'll finish it off.

Click on the 'Mission setup' button along the top left toolbar. Here is where you set the date, the time, the weather and type in the text for the mission Overview, Background and Intelligence. For this first mission we will use the default dates, times and weather. I think it's a worthwhile to get into the habit of writing a good overview. Maybe right now you are only writing missions for yourself, but you may find that you end up swapping them with friends or even posting them on the Internet. If you do that you want good, clear text to help the other players out. Remember, you know what's supposed to happen but nobody else does. In addition to a general setup of the mission the overview should always tell the pilot what they must do in order to successfully complete the mission. Here's a sample for this mission:

Overview: **You are flying a triangular grid, solo out of Henderson. Keep alert and destroy any enemy targets that you encounter. The mission will end when you have destroyed all targets of opportunity and returned safely to Henderson.**

Background: **Your plane was recently repaired and you want to take it out for a shakedown. You have planned a routine flight over Iron Bottom Sound.**

Intelligence: **Enemy activity has been at a low point lately. You should have an uneventful flight but watch for some Japanese patrol boat activity around Florida Island.**

Be creative and remember to have fun with it. Also remember, there is NO spellchecker in Mission Builder! Don't ruin a great mission with a sloppy Overview. When you're finished save your completed mission.

☛ I recommend that you never overwrite a mission when saving an updated version. Give each modified mission a new file name. For example, I might start with CC_01 for the first draft then add a letter each time, CC_01A, CC_01B, etc. until I get it tweaked just right. Then save your final version under whatever file name you like. I've found that overwriting a mission again and again can cause problems, especially with more complex missions.

Tips, Thoughts & Troubleshooting "Beginner's Luck"

Hopefully you enjoyed writing your first successful mission. You should have found that mission building is not difficult, but it is time consuming if you do it right and it does require that you pay attention to detail. Anyone can slap together a shoot-em-up with you against 50 Zeros, but if you want to put together a mission that makes sense and has a logical flow you need to give it a little advance thought.

Know what you want to do before you start. Depending on your personality you may want to make an outline, jot down some random notes, or simply visualize the entire mission in your head. Whatever works for you is the right thing to do. I maintain that you'll have more success if you think the mission through **before** you start Mission Builder rather than trying to make one up as you go along.

If the mission doesn't behave properly it is probably due to missing or inaccurate triggers or events. In this mission we could have written all four triggers at once then done the events. That may be a shortcut but it can get you confused. It's better to take one entire trigger/event sequence at a time until you become thoroughly familiar with mission building. Another thing you can do is to fly the mission after each trigger/event sequence to make sure you have that one correct before moving on.

If you fly the mission and did everything right and still fail then one of your goals is incorrect. Setting goals is helpful. Determine which planes, ships, vehicles, etc. you want to destroy as a part of the mission and label them as such when you add them. Don't inadvertently have "Must survive' as a mission goal for all of the enemy planes or you'll never succeed (unless you're a terrible shot). By the same token, every enemy plane doesn't have to have 'Must be destroyed' as a goal. If you go up against two flights of bombers and three packs of fighter escorts, that may be more planes than one person can realistically expect to shoot down. A plane labeled as 'None' under goals can either be shot down or not and it won't affect the mission outcome. I'll address some of these issues in later chapters.

Did you run out of ammo and find yourself flying around and around with no where to go? See the chapter "Avoiding The Never Ending Mission" on page 75 for help with this and other problems.

Use Mission Builder to open and examine the stock missions that came with CFS2. By studying the waypoints and reading the triggers/events you can learn a lot about constructing successful missions. Do the same thing with missions you download from the Internet. Add-on missions range in quality from excellent to terrible. Fly them first then dissect them. Again, it's a great way to learn not only the correct way but also the wrong way of writing missions. I don't hesitate to copy add-on missions and work at fixing the bugs and tweaking them more to my liking. It goes without saying that you shouldn't modify the stock missions (other than changing the plane to your personal favorite) without making back-up copies first. Experiment and be creative. Remember that you're doing this for enjoyment so have fun and don't get frustrated. In the next chapter we'll build on this lesson and add to our knowledge.

☛ You can save yourself a couple of keystrokes when entering waypoint numbers in triggers by simply typing the number without the plus (+) sign in front. Mission Builder will insert a plus sign for you. Positive waypoint numbers, those with a plus sign, denote waypoints along the Primary Route. By placing a minus sign (-) in front of your waypoint number you are indicating that it is a waypoint on your Alternate Route. We'll cover alternate routes later in the Handbook.

Chapter three – Raid on Tarawa

Preflight

In this chapter we will build on the lessons in Chapter Two and design a mission in which we conduct a bombing raid against Japanese ground targets, engage a few enemy fighters, assist a crippled bomber home and return safely to our base. Ready? Let's start.

Begin by clicking the New Mission button in the upper left-hand corner. This will bring up the window shown here. For this mission, click on 'Airbase', scroll down to **Tarawa – Bititu I**., highlight it, and then click on **OK**.

The map now places you at Tarawa, flying north in a Zero. Tarawa is going to be our objective. Before we change planes and lay in our waypoints, let's zoom in on the base and set up our targets.

First, place your cursor on top of waypoint #2 and right click. This brings up a pop-up menu. Select 'Delete' and left click. Now, drop down to the buttons along the bottom and left click on the plus sign (**+**), changing your cursor into a magnifying glass. Place the cursor directly on top of Tarawa and click. This will begin our zooming in on the island. Do this four times until your image looks like the one below.

As you look down on the island you'll notice that in addition to the landing strip, there are a number of other features already placed at the base. Over to the left we can see a wharf, while in the center are aircraft revetments and some buildings. Default infrastructure such as this, comes with each airfield and can be bombed and destroyed during a mission. However, you can't set a 'Must be destroyed' goal on the default infrastructure. We will take advantage of these existing structures when setting out additional targets. Place your cursor on top of the wharf and click four times to zoom in. This will give you a close up of the area surrounding the wharf.

Placing Infrastructure & Targets

We're going to place a Japanese vessel at the wharf. On the tool bar above the map press the **New infrastructure** button. It's the button that looks like a bull's-eye and cross hairs. Even though we are going to place a ship we do not use the 'New ship' button as we did in Chapter Two. That button is for *moving* ships. Since we are going to place a stationary ship it is considered to be infrastructure.

After clicking on 'New infrastructure' move to the left side of the screen, under 'Class' select **Ship** then scroll down to **HIJMS Tidori class Torpedo Boat** and select it.

Place your cursor at the wharf along side one of the docks and left click to set the torpedo boat, then right click to turn off the feature. Since the wharf is already oriented North (0°) you don't have to change the orientation on the boat. You may have to move the boat back and forth a little to get it just right. Move it by left clicking and dragging it where you want it. Make sure all of it is in the water when you are finished.

Go to 'Goal'. You have three choices here, 'Must survive', 'Must be destroyed', and 'None'. You'll recognize these as the same sets of goals we had for ships and aircraft in Chapter Two. If you select 'Must be destroyed' then destroying this object becomes a goal for the mission. If you fail to destroy this object, no matter how many other things you destroy or how well you fly the mission elsewhere, you cannot fully succeed. By the same token, if you select 'Must survive' you will fail the mission should that object be destroyed no matter how well you do on the other parts of the mission. By selecting 'None', the status of the object does not affect the mission outcome. You can destroy it if you like but not doing so won't count against you.

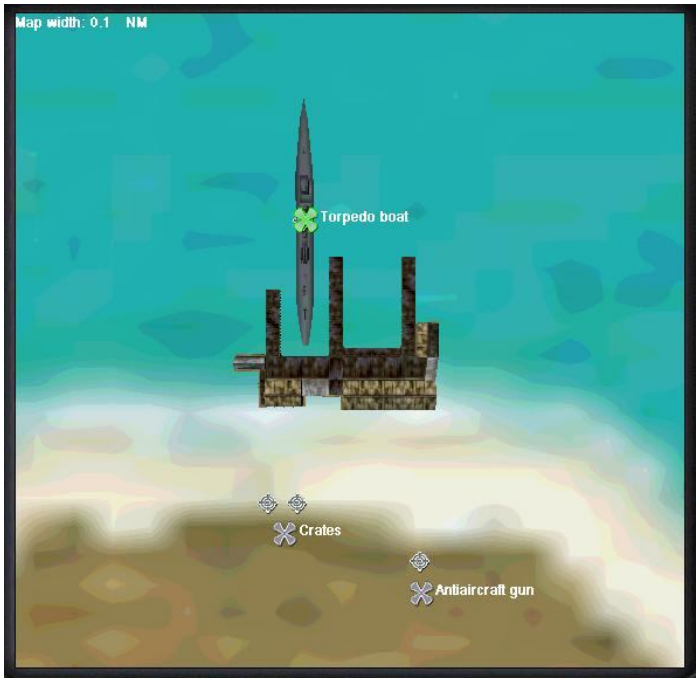
Most of the time you will probably select 'None'. This gives you more flexibility in flying the mission. Don't make the mistake of selecting 'Must survive' for all of the enemy infrastructure you place. I've seen this many times on downloaded missions. If your mission is to destroy an enemy fuel depot but the entire infrastructure placed there is marked 'Must survive' you will never successfully complete the mission unless you don't hit anything. It's your mission and you're free to design it any way that you want but give some thought to how to use goals to enhance the flying experience. For this mission set the goal of the torpedo boat as 'None'. You can destroy it if you'd like but it isn't necessary to complete the mission.

Now, click the ‘New infrastructure’ button again. This time select ‘Infrastructure’ instead of ‘Ship’ as the class. Select **Crates, Stacked** and place them above the beach, behind the wharf. Place two sets side by side. Right click, and then follow the same procedure to select a 20mm Japanese antiaircraft gun and place it along the shore line as well. Use the default goal of “None” for each of these items.

Be sure to pick a Japanese antiaircraft gun. You are going to fly this mission as an American. If you select an American antiaircraft gun it won’t fire at you so be sure it’s Japanese. When you’re done the wharf area should look something like the illustrations below.

I have labeled the infrastructure we added for you on this illustration. These labels will not appear on your screen.

Wharf before adding infrastructure



Wharf area completed

Now let’s move over to the center of the airfield and add some more targets. First zoom back out a little by clicking on the minus (-) sign twice. Then either click on the ‘Pan’ button (shaped like a hand) and, while holding the left mouse button down, drag the image to the left, or back out a couple of more clicks then zoom in on the center of the airfield. Either way you should end up with an image that looks like the one on the following page.

Notice that the stock airfield already has two hangers and a number of aircraft revetments located between the runways. We are going to use those existing revetments and add target aircraft to them.

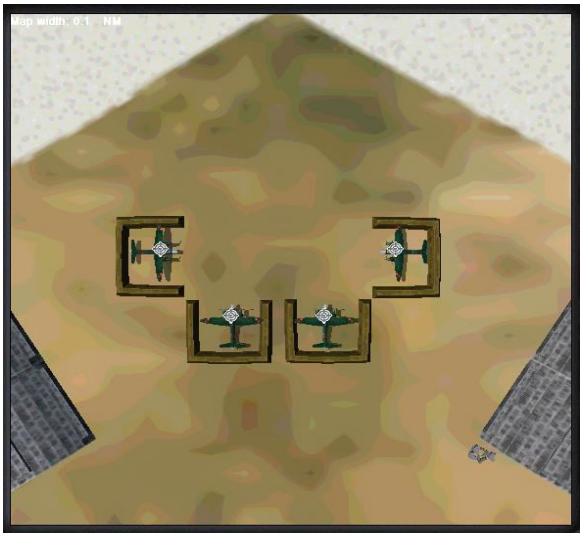
To make it easier to accurately place our planes, zoom in on the cluster of four revetments right in the center of the picture by changing your cursor to the plus (+) symbol and clicking twice more. Right click when you're done.

We are now going to place four Betty bombers. These are the primary targets for this mission. They will be stationary so click the 'New infrastructure' button and select **Aircraft** under 'Class'. Scroll down and select the **G4M2 Betty**. Set the goal to **Must be destroyed**. Place one plane in each revetment. Right click when you're done. You will need to set the goal for each plane.

We now have one bomber in each revetment. The bottom two are facing out but the two on the sides are facing into the walls. That's no good. We need to change the orientation on them to make them face out and look more realistic.

Orientation refers to the direction the item of infrastructure is facing in relation to North. Remember that a compass has 360 degrees. North is zero degrees (also 360°). If you face due East you are facing 90°. South is 180° and West is 270°. When you place infrastructure in a mission the default orientation is always 0° - North. With the four revetments we are using, two are facing North so we don't have a problem. With the other two we need to change the orientation. The Betty on the left needs to be facing East (90°) and the Betty on the right needs to be facing West (270°).

Place your cursor on the 'bull's-eye' of the Betty on the left and left click. This will turn the bull's-eye green. Go over to the infrastructure panel and type in **90** for the orientation and hit the 'Enter' key. Do the same thing for the Betty on the right, typing in **270**. This is a good time to make sure that the goal for all four bombers is set as 'Must be destroyed'. When you're finished the revetment area should look like this:



This is probably a good spot to talk about adding additional infrastructure to your missions. One of the fun aspects of Mission Builder is the ability to add extra objects to the existing scenery. This not only makes the area look more realistic but it gives us lots of extra targets to destroy and added antiaircraft guns make the missions more challenging.

That's all well and good, but be aware that for every piece of additional infrastructure you add to the game, you pay

a price in reduced performance. If you have a fairly new system you can probably pack your missions full of infrastructure without seeing any reduction in performance. If you have an old system (one that was new when CFS2 came out) you may find that the more items you add the lower your frame rates.

If you're seeing performance fall off as the amount of infrastructure increases you might want to ask yourself what the purpose of the extra infrastructure is. Will it be a target? Does it add to the excitement of the mission or to the mission goals? Or, is it just there to add realism to the background? If that forest of palm trees is important to you then by all means add it to your mission. Just recognize that you make a trade off in performance as the number of planes, ships, trucks, and trees, etc. climbs.

We've now added our main targets for this mission. Zoom out a couple of clicks and we'll add a few additional items of infrastructure to make it interesting. Following the instructions above, place two Val dive-bombers in any of the open revetments you choose – remember to get the orientation right. Also add a Japanese 75mm antiaircraft gun to the area between the two hangers. The Vals will make it tougher to pick out the correct targets and the antiaircraft gun will add an extra challenge. Set the goal for all three items as 'None'. When you're finished it will look something like this. The labels are just for clarity and will not appear on your screen.

☛ If you'd like, you can add even more infrastructure such as tents, buildings or additional planes and antiaircraft guns.

If you look closely you'll see that Tarawa comes with lots of empty gun emplacements. For an extra challenge, zoom in and add additional AA guns to the center of some of those emplacements.

Plotting Enemy Planes and Waypoints

Zoom out four clicks so that you can see the whole island. Click the 'New aircraft' button above the map. Select the **A6M2 Zero** as the plane, **Japan** as the nationality and **Intercept** as the mission type. Choose any call sign you like, the default is 'Abe'. Leave the payload as 'Guns only'. Select **2** as the number of planes. Set the skill level to **Veteran**, aggressiveness to **Average** and the goal to **None**.

Place the plane in the center of the island and left click, setting the first waypoint. Drag the cursor toward the lower left-hand corner of the map, a short distance into the darker water. Left click again, setting the second waypoint. Drag the cursor back to the island and left click once more setting the third waypoint. Now right click to stop setting waypoints.

Left click on the plane icon to select it. Go to the 'Route' tab on the left side of the screen. If your cursor was right on the plane when you clicked, it will be showing waypoint #1. If it doesn't, use the left and right arrows to get to waypoint #1. For 'Action' at waypoint #1 select **Take off**. Move to waypoint #2. Set the action as **Intercept all** and the altitude to **1500** feet. Move to waypoint #3 and set the action to **Land at base**.

Setting the Event Area

Event areas can be used to trigger events and control the action within a mission. You can have multiple event areas within a single mission and tie different triggers/events to each one. Also, event areas can overlap or a smaller one can be entirely within a larger one. In this mission we will use an event area to control warping while we are bombing our targets.

Click on the minus (-) sign one time to back out a little farther. Click on the **New event area** button on the tool bar. This is the button that looks like a circle with diagonal lines running through it. Place your cursor directly on top of the center of the island and left click. Holding the left mouse button down, drag the cursor downward. Notice that as you do this, a circle appears on your screen and grows larger or smaller depending on whether you are sliding your cursor up or down. Also note that at the bottom of the 'Event area' panel the size of your area is displayed as you drag. Adjust the size of the event area until it is **3** miles in radius then release the mouse button.

The event area panel has a place for the name of the area. Type in **Target Area** as the name. This panel also shows the exact longitude and latitude of the event area along with the size of the area. If you missed getting it right on 3 miles dragging your cursor, you can type in the exact radius here. Finally, click on **Apply** to add the name and correct the area radius.

After plotting the Zero's waypoints and setting the event area your screen should look like this:

Setting the Player Waypoints

Now let's set the waypoints for your plane. Zoom out two clicks. That should bring your plane into

view. Right click to clear the zoom feature then left click on your plane to highlight it. For your plane select the **F6F-3 Hellcat**, 'Nationality' is **USA**, 'Mission' will be **Strike**, for 'Unit ID' select any callsign you'd like. The 'Payload' will be **Bombs Medium**, '# of aircraft' equals **3** and the 'Formation' should be **Fingertip**. Set the 'Skill' level as **Veteran** and the 'Aggressiveness' as **High**. This is the skill level and aggressiveness of your wingmen. Your own ability determines your levels in these areas. For this mission we want good wingmen flying with us.

Place your cursor on top of your plane then holding the left mouse button down; drag your plane over toward the lower left-hand corner of the map. We will approach from the Southwest. Now with the cursor on top of your plane, right click. This brings up a pop-up window. Select **Insert** and a dot for waypoint #2 will appear. Drag waypoint #2 to the edge of the Combat Area. Right click on it and insert waypoint #3. Move it to the center of the island. Don't worry about exact location, we'll fine-tune it later.

Insert a fourth waypoint. Before you move it anywhere click the 'Maximum zoom out' button along the bottom row. Click the 'Pan' button and slowly move the map over to the right until Ocean Island comes into view. Now drag waypoint #4 down toward the center bottom of the map about the same elevation as Ocean Island. Then insert waypoint #5 and drag it to Ocean Island. Your map should look like this illustration:

Now let's fine-tune the waypoints. Click on your plane (waypoint #1). Then go to the 'Route' tab over on the left. Leave the action as 'Turn' but change the elevation to **2000** feet.

Click the right arrow and go to waypoint #2. You need to zoom in very close to fine tune this waypoint. Keep zooming in until the map width is at 2.6 miles. Adjust waypoint #2 until it is between a mile and a mile and a half **outside** of the Target Area. After you get it positioned just right, click on waypoint #2 again and set the elevation to **2000** feet. Click on the **Warp to** button.

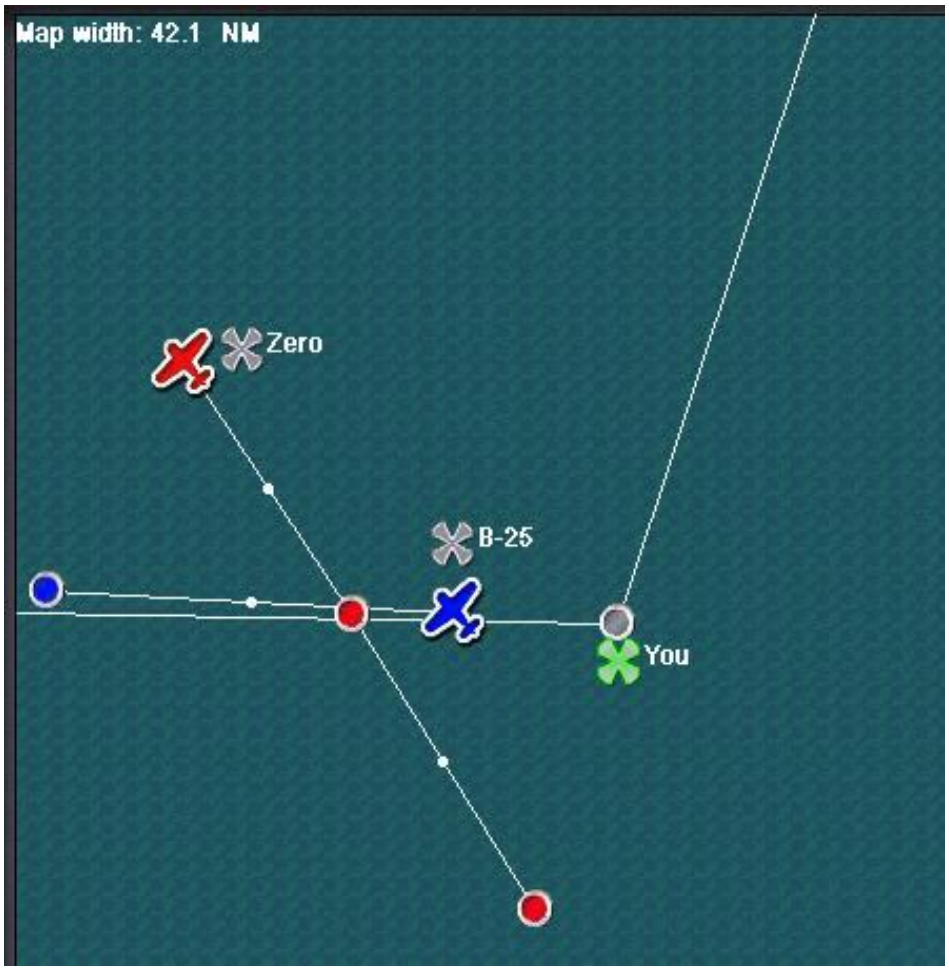
Stay zoomed in and move to waypoint #3. You may want to zoom in one more click. Move waypoint #3 directly over where the Betty's are parked. For 'Action' select either **Attack with bombs** or **Attack with all weapons**. As soon as you select this a 'Target' line will pop up and it should say 'G4M2 Betty'. If it doesn't your waypoint isn't directly over the bombers. Move it around until the target shows as the Bettys. Leave the altitude at the default setting and DO NOT click the 'Warp to' box. See the illustrations.

Waypoint #2 above, waypoint #3 below

☛ If you're having trouble seeing exactly where to place waypoint #3 because the Event Area is shaded too darkly, try this. Click on the **Chart Filters** button along the tool bar below the map. It looks like two circles overlapping each other. This opens a window that lists the different object types that are in your mission. You'll notice that there is an "X" next to "Event areas". Click on this "X" to remove it then click OK. The Event Area will disappear from the map. Don't worry; it's still a part of your mission. You've simply filtered it from view while you're working. To restore the Event Area to view, repeat the process, this time click the box next to "Event areas" to add the "X" back in. Filters are a handy tool to eliminate clutter when plotting complex missions with lots of objects.

Move to waypoint #4. Zoom out until the map width is about 21 miles. Leave the action as 'turn', set the altitude at 5000 feet and click the 'Warp to' box. Now we are going to add some more aircraft.

Click the 'New aircraft' button. Select a **B-25D Mitchell** as the plane. 'Nationality' – **USA**, 'Mission' – **Nothing**, 'Payload' – **Guns only** (he already dropped his bombs), '# of aircraft' – **1**, 'Skill level' – **Rookie**, 'Aggressiveness' – **Average**, 'Goal' – **Must survive** and set the 'Pre damage' to **20%**. Now you have a badly damaged bomber you are going to have to escort home. Place the first waypoint for the bomber 2 to 3 miles in front of your plane headed the same direction you are going. Place the second waypoint at the left edge of the map parallel to your course. Go to the Route tab and set the altitude at **4500** feet for both waypoints. Note that the action at waypoint #2 will be 'Land anywhere' that's OK for now.



Now add another new aircraft. This time select the **A6M2 Zero** along with the following parameters: **Japan**, **Intercept**, **Fujisawa**, **Guns only**, **2**, **Shotai**, **Veteran**, **Average**, **None**. Place the plane about 5 to 6 miles from the B-25 to the Northwest. Place the second waypoint directly on the bombers flight path and the third waypoint somewhere to the Southeast. The action around waypoint #4 should look like this: **Note: The planes are set farther apart here for clarity.**

Now go to the Route tab for the Zeros. Set the altitude at waypoint #1 at **5000** feet. At waypoint #2 set the altitude at **4500** feet and select **Intercept bombers** as the action. The default altitude and action are fine for waypoint #3 (they should never get that far).

After you have these two Zeros plotted, click the maximum zoom out button then pan a little to the left until Ocean Island comes into view. Drag the bomber's waypoint #2 over to the island and place it on top of yours. Now zoom in on Ocean Island, about 4 clicks.

If you have trouble picking out the B-25's waypoint marker from the cluster around waypoint #4 simply right click on the group of planes. A pop-up menu will appear listing everything that is located there and you can select the B-25.

Place your 5th waypoint right on top of the 'X' marking the airfield and on the route tab for waypoint #5 select **Land at base**. When you do this 'Ocean Island' should appear as the 'Base' name. Move the bomber's second waypoint anywhere over the island and leave it. The default action is already set at 'Land anywhere' which is fine.

Great job! You have all of the planes and waypoints plotted now let's tie it all together and give it some life with Triggers and Events.

Adding Triggers and Events

Let's get a mental picture of what we are doing in this mission. We are leading a flight of three Hellcats on a mission to bomb some Betty bombers parked at Tarawa. As we approach we are going to encounter some Japanese fighter cover. Once we have successfully dealt with the fighters and bombed the Bettys we will head home. On the way, we encounter a crippled B-25 that we need to escort back to the base. Once again we will tangle with more Japanese fighters. We need to design triggers and events to spawn the enemy planes at the correct time, enable warping and set saving the B-25 as a goal. This mission is more complex than *Beginner's Luck* was, so you need to follow the next steps very carefully. OK – let's do it.

The first thing we need to do is disable warp as we approach Tarawa and spawn the group of Zeros located there. Click on the 'Triggers' button and then on 'New'. Title this trigger **Player arrives at WP #2**. Select your plane if it doesn't show as the default. Drop down and for 'Type' choose **Status**, 'Param' will be **Waypoint reached**, the 'Operator' will be **Equals** and the 'Value' will be **2**. See below.

Click 'OK' & 'Close' then click on the 'Events' button. Start a new event and title it **Disable Warp – Spawn First Zeros**. Select **Player arrives at WP #2** as the trigger and click the double right arrows (**>>**). Under **Action type** scroll down and select **Miscellaneous** then select **Disable warp.** Next select **Spawning** then **Spawn unit**. "**Abe**" will appear in the **Unit ID** window as the default. Since that is the unit we want (Abe was the default name for the first Zero flight), click the right arrows then 'OK' and close the event window.

NOTE: You need to keep track of the various Unit ID's when setting events. We have two sets of Zeros, Abe and Fujisawa. If you spawned Fujisawa instead of Abe it would goof up your mission.

Our mission goal is to destroy the four Betty bombers. We MUST do that to fully succeed. We shouldn't leave the target Area (warp) until all four Bettys have been destroyed. However this is war and there is no guarantee that we will successfully destroy all four. We don't want our player stuck over Tarawa taking heavy fire and unable to leave. So, we will build in a time limit in which to complete the attack mission then order our Player to leave when the time is up.

Create a new trigger called **Player enters Target Area**. Your selections will be:

'Category' - 'Aircraft'
'Name' - 'Argo' (the player's aircraft)
'Type' - 'Status'
'Param' - 'Area'
'Operator' - 'Enters'
'Value' - 'Target Area'.

Close this trigger and create a new event. Title this event **Two Minutes in Target Area**. Select **Player enters Target Area** as your trigger. For your first Action Type select **Action delay**. Click the double arrows and a pop-up window will open where you can enter the amount of the delay you want in seconds. Enter **120** (60 x 2 = 120). Click **OK** to close the window. Next select **Miscellaneous** and then choose **Do nothing**. Click **OK** to save the event.

☛ I chose 2 minutes but you could make it any amount of time you'd like.

This is a Strike mission. You are to get in there and get out quickly. Even two minutes will seem like a long time with everyone shooting at you.

Remember, the purpose of this mission is to learn techniques, not recreate history.

We've disabled warp when we arrive at the target, now we need to let the Player know when it's time to leave and enable warp once he is safely outside the Target Area. Before we do that however, we are going to add one more factor to consider first. Remember, this is a Strike mission; we are carrying bombs. Regardless of how long the Player is over the target, we should not allow him to leave until he has dropped his bombs. So we will require the Player to complete his mission (drop his bombs) before we allow him to leave.

Create another trigger. Call this one **Player drops bombs**. Your selections will be:

'Category' - 'Aircraft'
'Name' - 'Argo' (the player's aircraft)
'Type' - 'Stores'
'Param' - 'Bomb count'
'Operator' - 'Equals'
'Value' - '0'.

OK, let's tell the player it's time to go. Create a new event entitled **Mission completed**. Select **Two minutes in Target Area** and **Player drops bombs** as your triggers (Note – we will have two triggers for this event). For your first Action Type select **Display Text**. Be careful here when you select Display Text. It is easy to select 'Debrief text' by mistake. Enter the following text: **Mission Competed – Turn to 199° and head for home**. Click **OK** then select Action Type: **Action delay** and enter **5** seconds. Enter another Display text of **Mission Competed – Turn to 199° and head for home** then another **5** second delay followed by a final Display text of **Mission Competed – Turn to 199° and head for home**. Finally, place your cursor in the box to the right marked "Enter description' and type in: **This event notifies the player that it is time to leave once he has completed the mission.** This description box is for you to enter notes about what an event is supposed to do. It can be helpful with events where the title isn't descriptive enough to tell you what the event does. When you're finished you will have created an event that flashes the message **Mission Competed – Turn to 199° and head for home** three times on the screen once the player has dropped his bombs and has been over the target for at least two minutes.

☛ How did I know to turn to 199°? Click on the Player's waypoint #3 (the triangle indicating the target) and then click on the Route Tab. Right above the Action window you'll see where it gives you the heading and distance to the next waypoint. Your actual heading may be a little different than 199° depending on where you set your waypoint #4.

The player has completed the strike mission and is headed out. Our next step is to enable warp. Create a new trigger called **Player leaves Target Area**. Your selections will be:

'Category' - 'Aircraft'
'Name' - 'Argo' (the player's aircraft)
'Type' - 'Status'
'Param' - 'Area'
'Operator' - 'Leaves'
'Value' - 'Target Area'

Close the trigger and create a new event. Title this one **Enable First Warp**. For your triggers select the Event **Mission Completed** and the Trigger **Player leaves Target Area**. Under **Miscellaneous** choose **Enable warp**. Click **OK** and close out the event window. See the following illustrations for an example of how your events should look.

Mission Completed event

Enable First Warp event

That completes our action over Tarawa. Now lets move on to waypoint #4 and the B-25 escort. Create a new trigger called **Player arrives at WP #4**. Your selections will be:

'Category' -	'Aircraft'
'Name' -	'Argo' (the player's aircraft)
'Type' -	'Status'
'Param' -	'Waypoint reached'
'Operator' -	'Equals'
'Value' -	'4'

Close it, and create another new event. Title this event **Spawn Additional Zeros**. Choose the trigger **Player arrives at WP #4** that you just created. You will use this trigger to disable warp for the player and spawn both the B-25 and the second flight of Zeros. First select **Miscellaneous** and choose **Disable warp**. Then select **Spawning** as the action type, then **Spawn unit**. From the drop-down menu select **Fujisawa** (the callsign for the second group of Zeros). Then select **Agate** from the drop-down menu and add it. "Agate" is the callsign for the B-25. You will end up with one

trigger/event spawning two sets of aircraft. Finally select **Display text** and type in the following: **Rescue that crippled bomber!** Click **OK** and close out the event window.

> ☛ Sometimes the order in which things happen in an event is important to its working properly. For example, in this event we want to be sure to disable the player's warp before we spawn any of the other planes. If you goof up the entry sequence you can use the **Up** and **Down** buttons below the "**In use**" window to change the order.

Next we'll enable warp so that we can jump back to the base when we're finished dealing with the Zeros. We'll start by creating a trigger entitled **Zeros destroyed**. Use the following parameters:

'Category' -	'Aircraft'
'Name' -	'Fujisawa" A6M2 Zero'
'Type' -	'Status'
'Param' -	'Object health'
'Operator' -	'Equals'
'Value' -	'0'

Make sure you choose the Fujisawa Zero and not the one we fought at Tarawa. Close the trigger and open a new event.

Title this event **Enable Second Warp** then choose both the event **Spawn Additional Zeros** and the trigger **Zeros destroyed** that you just created as your two triggers. Finally select **Miscellaneous** and choose **Enable warp** as your action and save the event.

Our situation here is very similar to what we faced in "Beginner's Luck". We must include the spawning of the second flight of Zeros with shooting them down in order for the event that enables warp to work right. Remember our three rules:

1. *An event will only occur once unless you check the "Repeat action" box.*
2. *An Object's Health equals '0' until it is spawned.*
3. *When an event is based on the destruction of a spawned object, always include spawning the object as one of the triggers***.**

If you are still having trouble understanding this go back up to Chapter Two and reread the sidebar "*Logic and Warping*".

Let's review how we are controlling warping up to this point. When we're over Tarawa we **don't** want to be able to warp until we finish our bombing mission. To insure that, we disable warp as we near the island and don't turn it on again until three things have happened:

- we drop our bombs
- we spend at least two minutes in the Target Area
- we leave the Target Area after completing the first two tasks

When we arrive at the B-25 we again disable warp. This time it is turned off until we shoot down the two attacking Zeros. Once they are destroyed warp is enabled. What is key about **this** warping event is that we must destroy the Fujisawa Zeros in order to warp home. You'll recall that we set the goal for both flights of Zeros as 'None'. Shooting them down is not a goal of the mission. You're ordered to leave Tarawa after two minutes whether or not you shot down the defenders. At waypoint #4 however, we must shoot down both planes to leave.

Notice that we handle warping very differently at waypoint #4 from what we did at the Target Area. There are many ways to control warping within a mission, only some of which will be covered in this Handbook. Experiment with different techniques so that you can match the best solution to the situation you are designing. If you are going to use warping to move the player then understanding how it works and firmly taking control of it is essential to a smoothly running mission.

☛ Writing missions isn't difficult but you do need to pay close attention to detail and the logical flow of events. The more complex the mission, the more potential for conflict between the various triggers and events that you use. Remember – in essence you are programming your computer to perform a certain function (run the mission); your programming must flow logically. Never assume that the computer (Mission Builder in this case) knows what you want it to do. You must spell things out slowly and clearly for it to understand. You do that by using clear and logical Triggers and Events.

OK, that's the meat of the mission now let's set a hidden goal and add a little frosting on the cake in the way of a pat on the back if we succeed. Open the triggers window again and create a new trigger titled **B-25 survives**. Then select the following parameters:

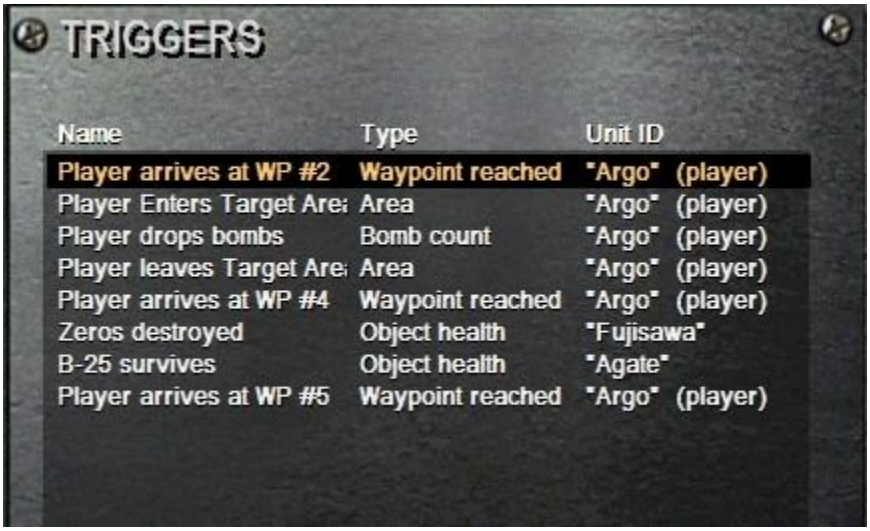
'Category' - 'Aircraft'
'Name' - 'Agate B-25D Mitchell'
'Type' - 'Status'
'Param' - 'Object health'
'Operator' - 'Greater than'
'Value' - '0'

Save it, and then create another trigger titled **Player arrives at WP #5**. Use the following:

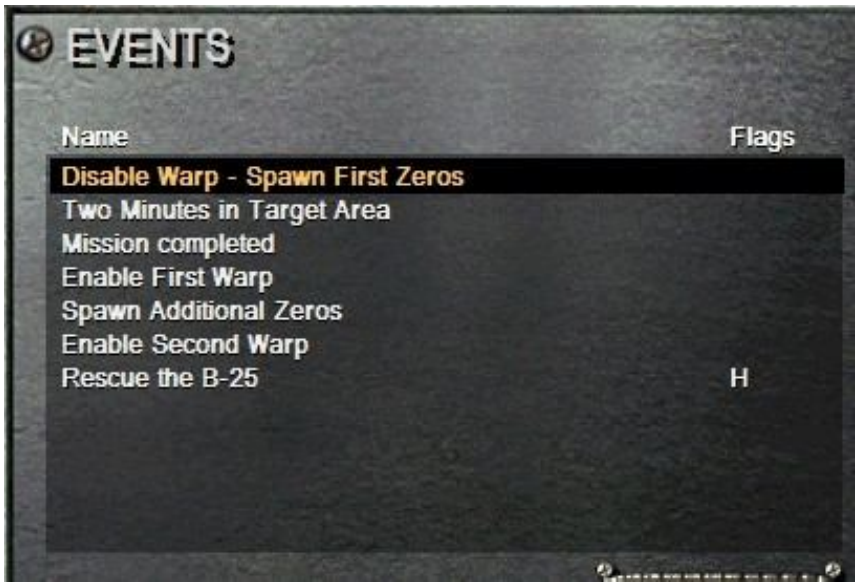
'Category' - 'Aircraft'
'Name' - 'Argo' (the player's aircraft)
'Type' - 'Status'
'Param' - 'Waypoint reached'
'Operator' - 'Equals'
'Value' - '5'

Save this trigger then open the event window. Create a new event titled **Rescue the B-25**. Select both **B-25 survives** and **Player arrives at WP #5** as the triggers associated with this event. The **Actions type** will be **Debrief text**. Click the right arrows and in the text window type in the following: **Great job! Your squadron inflicted heavy damage on the enemy bombers and rescued a crippled American B-25. The CO is impressed with your leadership and is recommending you for promotion.** Now click both the **Count as mission goal** and the **Hide goal from player** boxes. Saving the B-25 is now one of our goals but it won't show as a goal under 'Advanced info' prior to the flight. Of course you know it's a goal because you wrote the mission but anyone else who flies it (or you, if you forget) will have it sprung on them during the game.

You'll never see this message again unless you successfully complete all phases of the mission. If you do, you'll receive this nice little at-a-boy. Adding things such as specialized text like this gives a more polished look to your missions and is particularly nice for missions you plan to upload to the Internet.



Complete list of triggers



Complete list of events

That finishes all of the nuts and bolts of the mission. All we have left to add is the window dressing, i.e. the briefing and background information. We'll get to that in a minute. First, let's fly this thing and see if it really works! Before we fly, it would be wise to review what the mission is supposed to do. If you get it clear in your mind what is supposed to happen, then you will more easily recognize when something isn't working right and what, if anything needs to be tweaked.

The mission should begin with us leading a flight of Hellcats over the Pacific in route to bomb Tarawa. Our goal is to find and destroy the Betty bombers parked there. We will warp to a closer position. As we approach the island a flight of Zeros will take off to intercept us. Once we have dropped our bombs and spent two minutes in the target Area we will be ordered to leave. When we get about 3 miles from the island, warp will be enabled and we can jump to our next waypoint headed back to our base. When we reach that spot we will encounter a badly damaged B-25 that is about to be attacked by two Zeros. We need to shoot down both Zeros and escort the B-25 safely home.

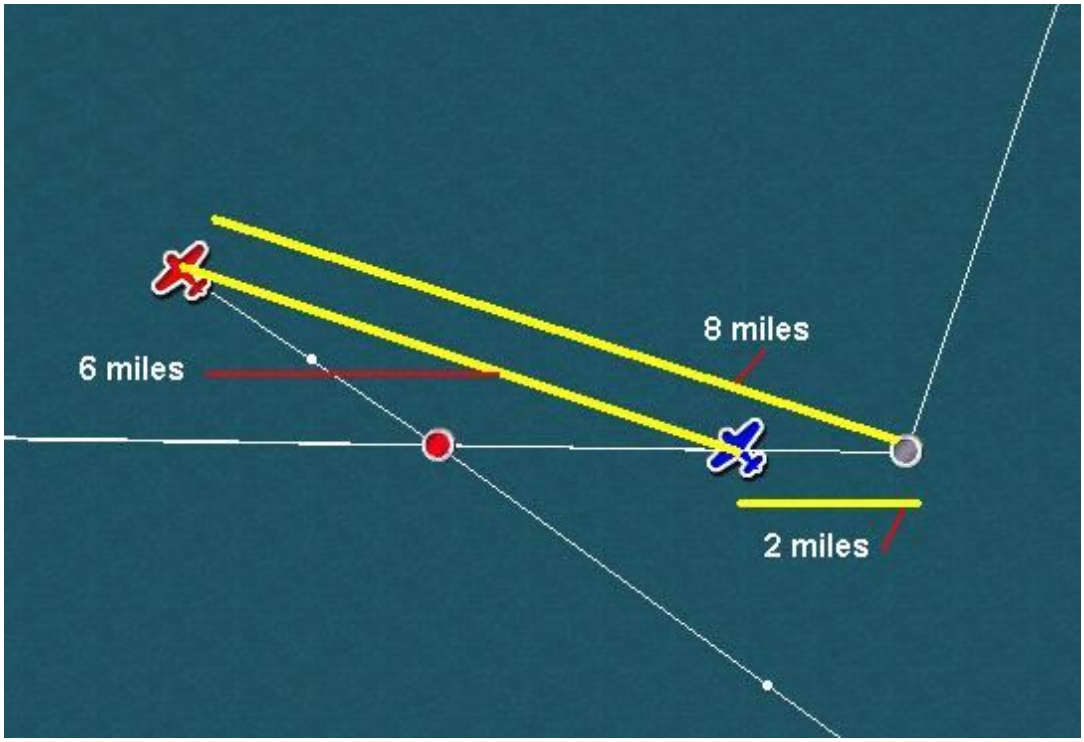
Now, let's go flying. Click on the 'Fly now' button. A window will open allowing you to save the mission. Make the mission name **Raid on Tarawa**. This is the title that will appear on the mission menu. The file name is the name the mission is stored under in the computer. Call it anything you like. Since this is our second mission in this series, I'll name mine CC_02. Once saved, you can fly the mission. Have a crack at it then meet me back here.

Well how did you do? Were you able to avoid the flak and successfully complete the mission? Did you notice that the torpedo boat we placed there was firing at you as well? Did you succeed or were you shot down over and over again? Here's a little tip. Before test flying a new mission that you've written go into 'Settings' and look under the realism settings. Change the player's plane to 'Invincible'. I would never play the game at this level but for testing missions it is very helpful. Even if you're an Ace you run the risk of a lucky flak shot or an accidental collision. It can get very frustrating to keep getting shot down before you can fully test the mission. Switching to invincible will solve that for you.

Did you notice the radio chatter? As in our first mission, *Beginners Luck,* the game automatically adds a number of extras, such as canned radio traffic, to the missions. It adds a little realism and keeps us from having to plug it in each time.

You may have noticed that even if you didn't destroy the Betty bombers you still got congratulations on a job well done. Remember that our debrief text (which congratulates you) is tied to your arriving at waypoint #5 and having the B-25 survive. It is not linked in any way to the Bettys. In Chapter Seven I'll discuss using multiple debrief texts to cover different outcomes.

You'll probably need to do some tweaking, especially around waypoint #4. It's a little tricky to get just the right balance between the B-25, the Zeros and you. You want to arrive at the B-25's location at the same time or just slightly ahead of the Zeros. If they get there first it's very easy for them to shoot down the B-25 because of the damage it has already suffered. If they do that you can't successfully complete all of the mission goals. Remember that the B-25 is flying away from you and toward the Zeros. The distances shown in this illustration are approximate but are one combination that seems to work. Work with it and adjust it until the mission flies the way you want it to.



OK, let's finish up by plugging in the narrative information. Click on the 'Mission setup' button. In the briefing 'Overview' window type in the following: **You will lead a flight of Hellcats on a bombing raid against the Japanese airbase on Tarawa. Your mission is to destroy the Betty bombers parked there and return safely to your base. This is a strike mission; get in, drop your bombs, and get out. If you did not destroy the bombers on your first pass come back and hit them again. LEAVE THE AREA WHEN ORDERED TO DO SO.**

Under 'Background' type in: **The Japanese bomber missions launched from Tarawa have been wreaking havoc on our airbases and convoys. If you can destroy the Bettys based there it will buy valuable time for our troops.**

Your 'Intelligence' information will be: **The Japanese have followed a steady pattern with their bombing raids. Based on this routine, G2 estimates that today's raid stands an excellent chance of catching their entire bomber fleet on the ground. FOLLOW YOUR ON-SCREEN INSTRUCTIONS. LEAVE TARAWA WHEN ORDERED TO DO SO.**

Remember, when you write these narratives you're not writing them for yourself. Write them as though they were going to be read by someone who knows nothing about the mission. This is especially important for missions like this one where you want to player to follow on-screen instructions. Don't hesitate to emphasize your instructions in the briefing. Additionally, your narrative should make sense, help build the story line, and be grammatically correct. There is no spell check in CFS2 so be careful!

The default date of June 1, 1943 is fine. The U.S. Marines didn't land on Tarawa until November 20th of that year so any date prior to that is fine. (Although it really doesn't matter, I have made no attempt to be historically accurate in this mission. This is after all a tutorial on mission building, not a history lesson.) Set the time for 09:00. We'll strike in the morning and have all day to get home. You can set the weather to anything you'd like. Remember though, more clouds may mean decreased performance.

That's it! We're done and you've now written your second successful mission!

Tips and Thoughts

Raid on Tarawa is more complex than *Beginner's Luck* but the basic concept and principles remain the same. We merely added some additional features and more complex triggers and events. I hope you're learning that Mission Builder is a powerful tool that gives you a wide range of options in designing your missions. Here are some thoughts and tips regarding *Raid on Tarawa.*

Mission Too Hard:

- eliminate some or all of the antiaircraft guns you added
- reduce the number of Bettys from 4 to 2, or
- move a couple of the Bettys onto the runway where they are easier to hit (Don't place them on the runway the Zeros are using or they may collide)
- reduce the skill level of the Zeros to 'Rookie'

Mission Too Easy:

- add more antiaircraft guns
- spawn more Zeros – spawn flights of 6 instead of flights of 2
- add another waypoint with more enemy fighters
- raise the skill level of the Zeros to 'Ace'

You can also do any number of other modifications to the mission to suit your tastes. For example, you can replace the 'infrastructure' torpedo boat with a moving one and lay out waypoints so that the torpedo boat circles the island while you are bombing similar to the way you set a moving torpedo boat in *Beginner's Luck*. This will give you antiaircraft fire that is constantly shifting position.

If you'd prefer to bomb a convoy instead of ground based targets simply eliminate Tarawa all together. Make your second and third waypoints out in the open ocean. Instead of Bettys place Japanese ships sailing in formation. You can still spawn a flight of Zeros, just have them take off

from a carrier instead of the island. In other words, design your missions any way that you want as long as it's fun for you. Use your imagination and be creative.

Troubleshooting Raid on Tarawa

You have three warps in this mission. The first one is automatic and should cause you no trouble as long as you checked the 'Warp to' box for the second waypoint. If you have trouble warping it will probably be at the other two locations.

First, be sure that you checked 'Warp to' on waypoints 2 & 4. That is the easiest fix. If that is OK then the problem lies with your triggers and events. Go back over the section on triggers step by step. We created an event to disable warp when we reach waypoint #2. We keep it disabled until three things occur: a) we drop our bombs, b) we're inside the target Area for at least two minutes, and c) we leave the Target Area. We MUST turn warp back on when we leave the Target Area for it to work again.

If you warp after leaving the Target Area but end up at Ocean Island instead of Waypoint #4, then you forgot to mark Waypoint #4 as "warp to". If you get to Waypoint #4 safely but can't warp home then you either forgot to enable warp after the second flight of Zeros were destroyed or you didn't destroy both Zeros. Remember, at both Waypoint #2 and #4 we first turn off warping then turn it on again after our conditions have been met. **That is always the best way to handle warp – turn it off then turn it back on again after your conditions have been met**.

If enemy planes are in the area the game automatically disables warping, but don't rely on this feature. You should always turn warping off and on using triggers and events. If you enable warp and there are still enemy planes close by you won't be able to warp until they are destroyed or you get far enough away from them that the game releases warping. That's fine; at least when you control warping you know that it will work as you planned once the skies are clear.

If all else fails, in this or any other mission with lots of things happening, you can always delete all of the triggers and events (you must delete the event first) then start over. It's not as hard as it sounds, especially if you've thought out your mission thoroughly or if you've made notes on what you want to have happen. This is good to remember because if you have a big mission with lots of infrastructure, planes, ships and waypoints you don't have to rewrite the whole thing. Just delete the triggers/events and try again.

You may find after warping from waypoint #4 that instead of approaching your base and ending the mission you are over empty ocean with the blue "Press 'X'…" message and an indication that your engine is sputtering. That happens because you are out of fuel. You need to do one of three things, fly unlimited fuel (not very realistic), move waypoint #4 closer to Ocean Island, or be more conservative on your fuel usage.

You will probably notice that your wingmen are very ineffective when it comes to bombing the Bettys. As soon as they see the Zeros they get distracted and forget all about bombing. In my experience this seems to be a bug with the AI aircraft (AI = Artificial Intelligence). I can successfully design bombing missions where the AI bombers hit their targets but I have much more difficulty getting my wingmen to do the same. Even when using settings of "Ace" and high aggressiveness with no enemy planes around they still want to fly over without bombing. I have found that I get the best performance from my wingmen if I use my keyboard commands to bracket the target then give them the "Attack" command. Although they sometimes make an initial pass without attacking, given time they will usually return and bomb or strafe the target.

So, use their behavior to your advantage when flying the mission. As you approach Tarawa order your wingmen to attack. They will go after the Zeros freeing you up to hit the Bettys. With luck they will destroy one or both of the Zeros for you. If they shoot them both down, order them to rejoin you. Even if they don't drop their bombs on the Bettys they will draw off some of the antiaircraft fire. You definitely want to order them to attack when you reach the B-25 as you need all the help you can get keeping the Zeros off the back of the bomber.

Summary

By now you should be feeling more comfortable with the Mission Builder. While *Raid on Tarawa* is a little more complex than *Beginner's Luck* I hope you didn't find it difficult. The more comfortable you become with using triggers and events the more creative and challenging your missions will be. Experiment and be imaginative. Remember that you're doing this for enjoyment so have fun and don't get frustrated.

In Chapter Four we'll design a shoot-em-up. You know the kind of mission I mean; you fly your Corsair against the entire Imperial Japanese Air Force. Well, ours won't be quite that bad but there will be lots of planes, lots of action and lots of choices for you to make plus multiple ways to end the mission.

Multiple Installs

You may find that as your collection of add-on CFS2 planes, ships, and scenery grows it becomes too cumbersome for a single installation. You can easily do multiple installs of CFS2 so that you can have one for each theater or time frame that you're interested in. I've tried various ways of doing multiple installs but this is the method that I like best. It's easy and never fails to work:

1. If you did a standard installation, your game is located in a folder named **Combat Flight Simulator 2**. Start by renaming this folder to something different, such as **CFS2 Pacific**.
2. Use your original installation disks and simply install another version of the game. As long as the installation program doesn't see a folder named **Combat Flight Simulator 2** you can install a new version from the disks.
3. You can then leave the name of this new install as is or you can change it to something different such as **CFS2 Europe**.
4. Keep following these steps for each additional installation that you want to have.
5. If you had any shortcuts on your Desktop or elsewhere to the original installation you will need to change the path on those shortcuts to reflect the new folder name. You can edit the shortcut properties using Notepad, but there is a simpler way:
    a. Delete all of your old shortcuts
    b. Open your new folder (*CFS2 Pacific* for example), Right-Click on the CFS2 icon (the pilot's face) and select **Send To – Desktop (create shortcut).** Do this for both the game and Mission Builder icons.
    c. Rename your new Desktop shortcuts to whatever you like and you're in business.

Chapter four – a “wake” up call

Preflight

For this lesson we are going to write what I call a shoot-em-up, a mission with lots of planes and lots of action. We will take off from a carrier located near Wake Island. Both the fleet and the base are coming under attack and we must engage the enemy and try to defend them both. In this mission we will be using delays to prolong the action and complex triggers and events to end the mission. We will accomplish all of this without using any warping. That will be good news for some of you. Fire up Mission Builder and let’s get started.

Begin by clicking the New Mission button. Select ‘Airbase’, scroll down to ‘Wake’, highlight it, and then click on ‘OK’.

The map now places you at Wake Island, flying North in a Zero. All of our action will take place on and around Wake Island. Before we change planes, add enemy bombers and lay in our waypoints, let’s zoom in on the base and set our infrastructure and ships.

First, place your cursor on top of waypoint #2 and right click. Select **Delete** from the pop-up menu and left click. Now, drop down to the buttons along the bottom and left click on the plus sign (**+**). Place the cursor directly on top of Wake Island and click. Do this twice to zoom in on the island.

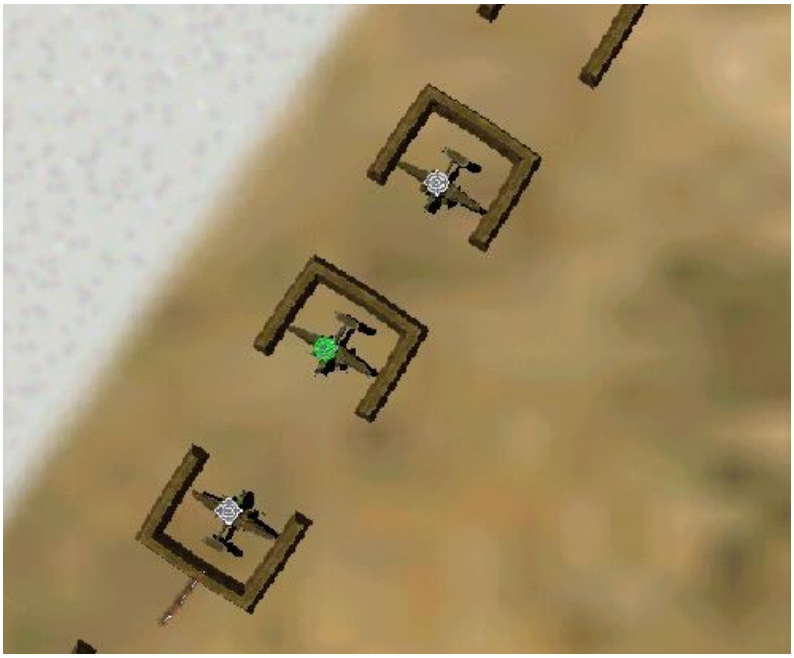
Notice that Wake Island is shaped like an elongated ‘V’ with the open end pointing northwest. There is a sheltered lagoon formed by the arms of the ‘V’ and open ocean surrounding the island. Now place your cursor on top of the airfield and zoom in three more clicks.

As you look down on the island, notice the additional infrastructure that’s placed there. We can see hangars, aircraft revetments, buildings and gun emplacements among other things. We will be using some of these items as targets for the Japanese bombers. Let’s add a few additional items to make it interesting.

Placing Infrastructure & Targets

Center your cursor on the aircraft revetments and zoom in three times. **Remember:** right click to turn off the zoom, or any other feature. On the toolbar above the map press the **New infrastructure** button. Under ‘Class’ select **Aircraft**. Under ‘Type’ scroll

down and select the **B-25D Mitchell**. Place 3 Mitchell bombers in any of the revetments that you wish. You will need to set the orientation on each one to get them pointed correctly. Set the orientation to 210° for bombers placed in the three upper revetments and 30° for those placed in the lower set. Leave the 'Goal' as 'None' for all three bombers. See the detail below.



B-25 detail

Now zoom back out to the 0.5-mile width (2 clicks). We are now going to place a wharf along the shore of the lagoon. At this level of detail you can just see the lagoon toward the upper left of your map. Zoom in 2 clicks on the shoreline along the lagoon.

Note: If you prefer you can move around the map area by panning using the button that looks like a hand. Click on the 'Pan' button then, while holding the left mouse button down, drag the map to get the area you want into the image window. Use whichever method (panning or zooming) you prefer for moving around the map.

Click on the **New infrastructure** button again and select **Infrastructure** under 'Class'. Scroll down and select the **wharf**, then place it along the beach so that the docks extend into the water. Be sure to get the orientation correct. In my example below, I have the orientation set at 0° as I have placed the wharf at a bend in the shoreline. You may set your wharf slightly differently.

While we're zoomed in we're going to add two ships. Both of these ships will be stationary so we will use the 'New infrastructure' button and not the 'New ship' button. Following the steps above and using **Ship** as the 'Class', select a **USS PT Boat** and place the PT boat at the wharf along side one of the docks. Then select a **USS Tanker Ship** and set it off shore a short distance from the wharf. Set the goal on the wharf and both ships to 'None'. See the illustration.

I have the PT boat oriented to 0° and the tanker to 230°. Your settings may vary slightly. When you're finished, zoom back out to the 0.9-mile map width.

Using the Pan button, move the map around until you have a good view of the built up area around the airfield. If

you look carefully you will see quite a few gun emplacements located around the island. These emplacements are empty right now. We will place antiaircraft guns in several of them at various spots around the island. In the illustration below I have circled some of them in red.

To place an antiaircraft gun, zoom in close on one of the gun emplacements. You will want to be at either the 0.1 or 0.0 map width. Click the **New infrastructure** button then select **Infrastructure** under 'Class'. Scroll down to where the antiaircraft guns are located. You have two choices here, the .50 cal machine gun and the 40mm cannon. Choose whichever one you prefer.

Note: You must select an American gun. If you select a Japanese gun it will fire at you instead of the Japanese bombers. In this mission Wake is an American held island so you must select an American gun to defend it. I have selected the 40mm gun in my example.

Orient the gun so that it is pointed away from the entrance.

Go around the island and place as many antiaircraft guns as you'd like, wherever you choose. You should place at least four to cover the four 'corners' of the island.

Note: Remember the lesson from Chapter Three, the more infrastructure you add to a mission the more drain there will be on your computer. This will be a very busy mission with lots of planes and ships so you may not want to over do it on the antiaircraft guns.

When you are finished placing your guns, zoom out to the 6.9-mile map width and using the pan function, move the Island down toward the lower left-hand corner of the map. See the illustration.

I've marked on this map the area we are going to focus on now, about 6 miles NE of Wake.

Placing the Fleet

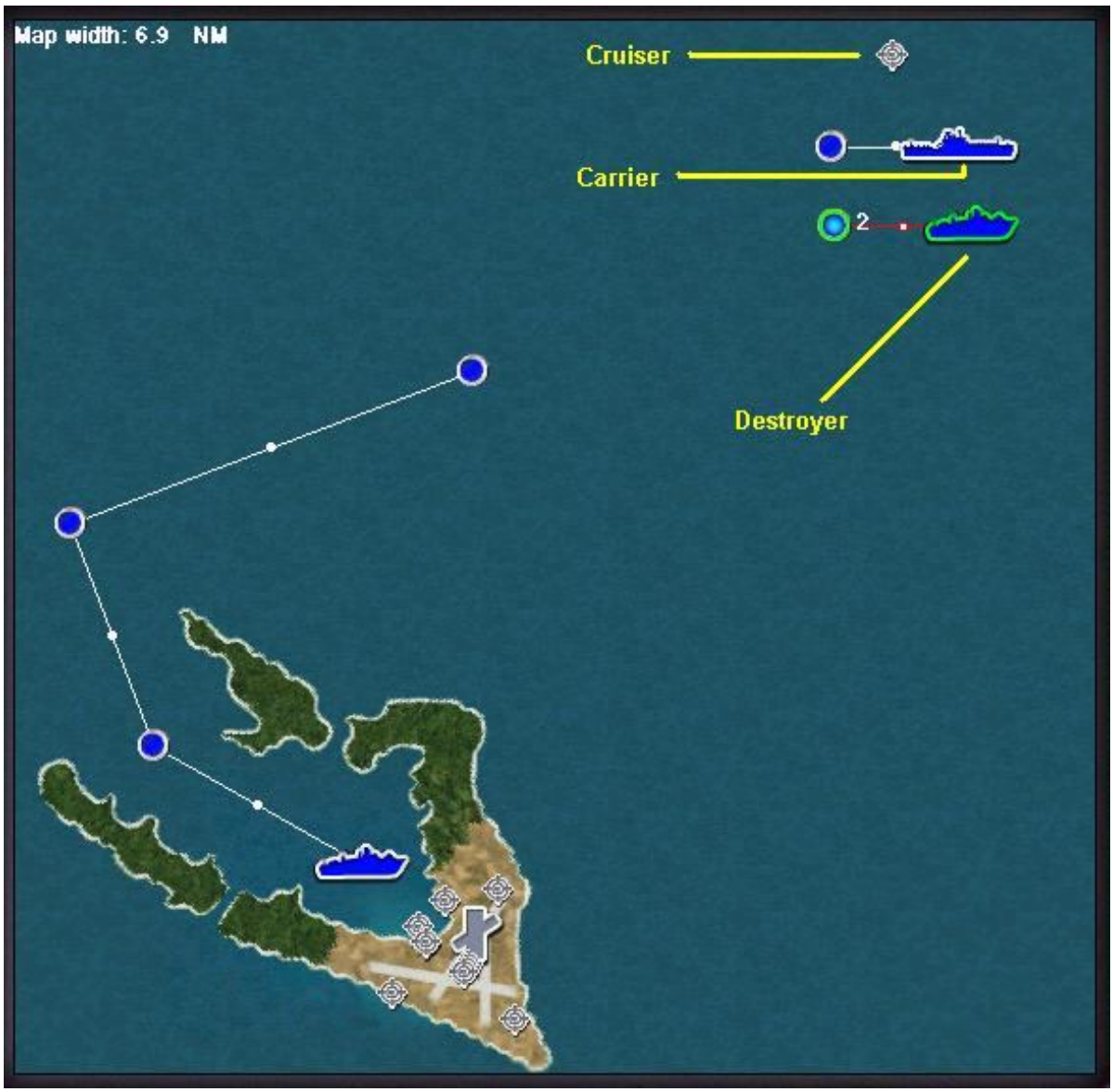
We are now going to place our aircraft carrier and a couple of support ships. Click on the **New Ship** button along the top row. It's the button that has a ship silhouette. Under 'Ship Type' select **USS Essex class Aircraft Carrier**. Set the 'Nationality' to **USA**. Select any unit ID you'd like. I chose 'Essex CV-9'. Leave the 'Skill' as **Novice** and set the 'Goal' as **Must survive** leave all other defaults as is. Place the carrier approximately 6 miles NE of Wake Island. Move the second waypoint a short distance to the West of the ship and right click.

Now select another new ship. This time make it a **USS Mahan class Destroyer**. The default nationality should already be **USA**, leave the other defaults as is. Place this ship next to the Essex and plot waypoint #2 as you did with the carrier.

Next click **New infrastructure** and choose **Ship** as 'Class'. Select a **USS Portland class Cruiser** and set the orientation to 270°. Leave the 'Goal' as **None**. Place this cruiser next to the other two ships. We now have three ships, two moving, and one stationary, located in the ocean a few miles Northeast of Wake. Before we fine-tune them, we'll add one more ship.

Click the **New ship** button again. Select another **USS Mahan class Destroyer**. Double check that the nationality is USA and set the 'Goal' to **Must survive**; the remaining default settings will be fine. Place this ship near the bottom of the 'V' formed by Wake Island. Drag waypoint #2 to the mouth of the 'V'. Then set waypoint #3 a little to the Northwest and finally set waypoint # 4 about midway between waypoint #3 and the fleet. Refer to the illustration for assistance.

We now need to fine-tune the waypoints and locations of our ships. Since we are working on the second destroyer we will start there. Highlight the second destroyer and click on the 'Route' tab on the left side of the screen. If you clicked on the destroyer to highlight it then you will be sitting on waypoint #1. If you are on any other waypoint, click the left arrow to get to waypoint #1.



Under 'Action' notice that you have two choices, 'Turn' or 'Do not evade'. You may have heard, or experienced, the fact that ships in CFS2 sometimes 'sail' overland. There is a bug in the game that will cause a ship to travel over land as though it was on water (see Chapter Nine). We would like to prevent that from happening if possible. Set your action to **Do not evade**. There is no guarantee that this will keep your ship from turning into the island but it should help. Set the speed at waypoint #1 to **3** knots. Move to waypoint #2 by clicking the right pointing arrow or on the waypoint marker. Keep the action as **Do not evade** and up the speed to **8** knots. At waypoint #3, set it to **Do not evade** and **12** knots and at waypoint #4, the defaults of 'Turn' and 16 knots will be fine.

Visualize what we've created here and you will see that when the attack begins this destroyer is stationary in the lagoon. She needs to get up to speed, which takes a while, and negotiate the narrow entrance of the lagoon to get in position to assist the fleet.

Now, zoom in on the fleet. If you set the ships close, you should be able to get all three in the frame at the 1.7-mile width setting. Adjust the positions of all three ships until they are about a half a mile apart. I set mine with the destroyer farthest South and the cruiser farthest North. For both the destroyer and the carrier set the 'Action' at both waypoints to **Do not evade**. Set the speeds as **1**

knot at waypoint #1 and **5** knots at waypoint #2 for both. You don't need to do anything with the cruiser, as it is stationary infrastructure and not a moving vessel. With all four ships in place and the waypoints plotted your map should look something like the illustration on the previous page.

Plotting the Enemy Attack

In this mission the Japanese will attack both the ships at sea and the island base simultaneously. Your mission will be to try and fend off both attacks as best you can. Both your carrier and the destroyer in the lagoon must survive to successfully complete the mission. Let's lay in the enemy attack on the carrier first.

Zoom out a couple of clicks to around the 20-mile width. Click on the **New aircraft** button. From the drop-down menu select the **B5N2 Kate**. Set the 'Nationality' to **Japan**, the 'Mission' to **Anti-ship**, 'Unit ID' to **Abe**, 'Payload' to **Torpedo**, '# of aircraft' to **3**, 'Formation' to **Fingertip**, 'Skill' to **Veteran**, 'Aggressiveness' to **Average** and the 'Goal' to **None**. See the illustration to the right.

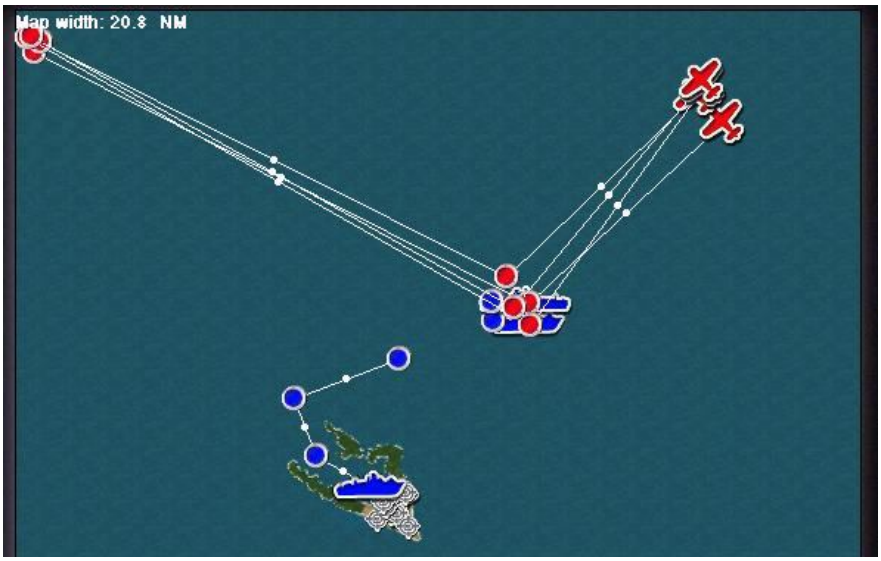
Place the first waypoint for this flight about 5 miles northeast of the carrier. Drag the second waypoint and place it on top of the carrier. Drag the third waypoint to the upper left-hand corner of the map. Now, click the **New aircraft** button again. You will keep the same parameters as before. Everything will be filled in for you but you must reset the payload to **Torpedo**. Place a second flight of Kates along the same flight path, changing only the second waypoint. This time place the second waypoint on top of the destroyer. Click the **New aircraft**' button again and create a third flight of Kates. Make the Unit ID for this flight **Otani**, set the payload to torpedoes and the mission to **Anti-ship**. Lay its course directly on top of the first flight, again targeting the carrier. Click on **New aircraft** and this time select the **D3A1 Val**'. Set the payload to **Bombs Heavy** and change the mission to **Strike**. **We use "Anti-ship" for moving ships and "Strike" for infrastructure ships**. Place the Vals along the same flight path as the Kates but target the cruiser as your second waypoint. You have now roughly plotted three waves of torpedo bombers and one wave of dive-bombers attacking the fleet. Now let's give them a fighter escort.

☛ When you're laying in multiple groups of planes that follow the same basic flight path, instead of creating a new plane each time and plotting the waypoints you can copy an existing plane and just fine tune the route. Here's how.

Click on any waypoint of the flight you want to copy to select it. Hold down the **CTRL** key while you **left-click** the waypoint marker and drag the flight slightly to one side. This will create a copy of the original flight.

The copy will contain all of the original information, such as aircraft and mission type, nationality, payload, etc. Only the Unit ID will be blank. Since you dragged the copy to a new position you must fine tune the individual waypoints.

This is a fast and easy way to add additional flights and works great when wanting to build large bomber formations in a hurry! ☺



Click the **New aircraft** button once more. Select the **A6M2 Zero** as the plane and **Intercept** as the mission type. If you have followed this sequence, the remaining information will be the same parameters as we used for the bombers. The payload will have defaulted back to 'Guns only' which is correct for the Zeros. Place these Zeros along the same flight path as the bombers with the second waypoint set anywhere in the cluster of ships. Your map should look something like this illustration.

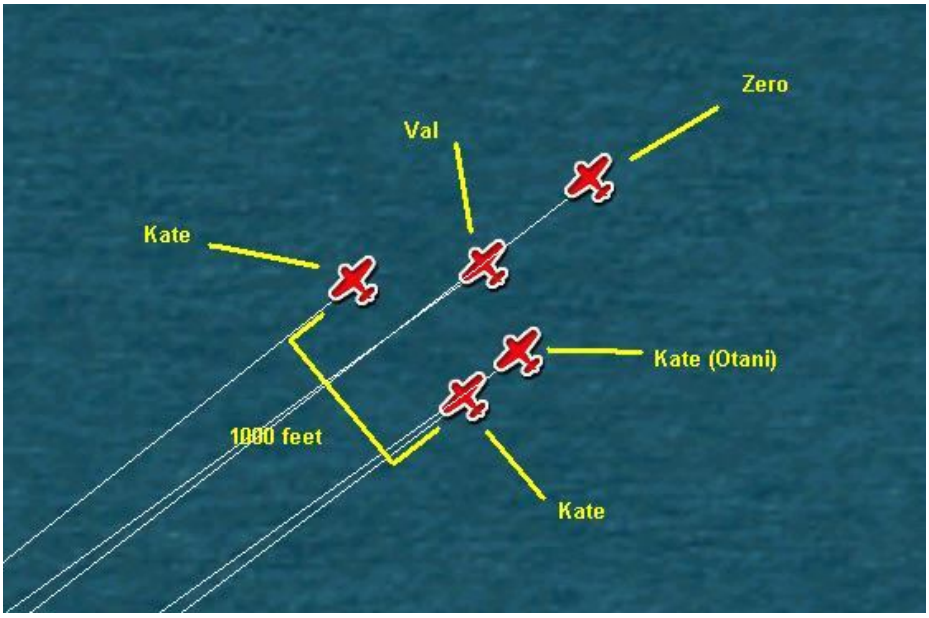
We now need to fine-tune the waypoints for this attack force. There are several ways to do this. I'll try to show you a method that minimizes the amount of zooming in and out you will need to do.

Start by separating the five airplane icons at waypoint #1 so that you can easily tell one from another. When you have them separated, highlight the first flight of Kates, the flight attacking the carrier with the unit ID of 'Abe', and go to the 'Route' tab. If you are not on waypoint #1, use the left and right arrows to get there. Set the 'Altitude' to **500** feet. The default action of 'Turn' is correct. Click the right arrow and go to waypoint #2. Here, set the 'Action' to **Attack w/torpedoes** and the altitude to **500** feet. The 'Target' may show as 'Current point', that's OK for now, we will fine-tune that later. Go to waypoint #3 and set the altitude at **3000** feet.

Highlight the second and third flight of Kates. Following the steps above set the parameters the same as the first flight with the altitude at 500 feet for both waypoints one and two, 3000 feet at waypoint #3 and the action at waypoint two as 'Attack w/torpedoes'.

Highlight the flight of Vals. As dive-bombers we need to adjust their altitude some. Set their altitude as **1500** feet for waypoints one and two and **3000** feet for waypoint #3. Then, set their airspeed to **157** knots at waypoints #1 and #2 to match the speed of the Kates. Set the action as **Attack w/bombs** at waypoint #2. Finally highlight the Zeros. Set their altitude as **2000** feet at waypoints one and two and **3000** feet at waypoint #3. Set the action at waypoint #2 to **Intercept all** and, as with the Vals, reduce their speed at waypoints #1 and #2 to **157** knots.

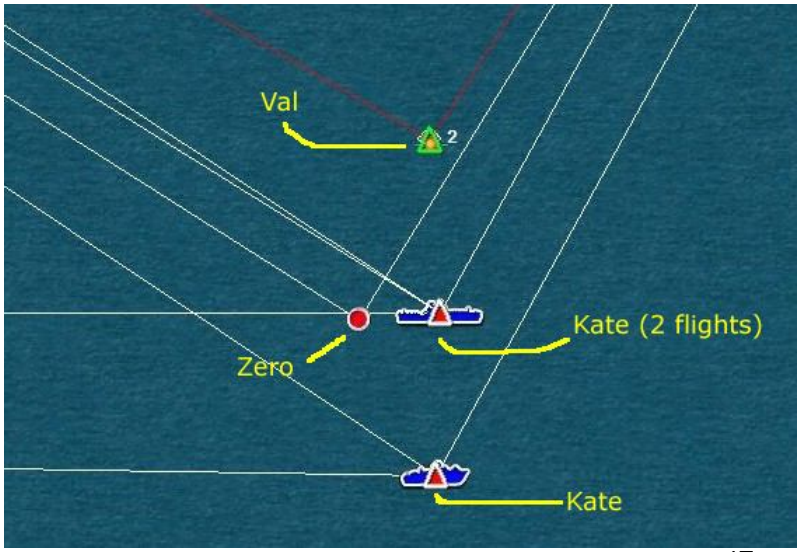
We want all of the planes to arrive at waypoint #2 at about the same time so we need to adjust the speeds to match that of the slowest plane in our group, the Kates in this case. We could set the speeds the same at waypoint #2 also but I have chosen not to do that. This method will create some spacing as they approach waypoint #2 and the speeds of the Vals and Zeros pick up. The Zeros should get there slightly ahead of the bombers to engage our planes.



You have now set the altitudes and actions for the first wave of enemy bombers. We next need to zero in on each waypoint to tighten up the formations and specify the targets. First, place all of the plane icons on top of each other at waypoint #1 then zoom in about 5 clicks to around the half-mile width. Set the five flights so that they are reasonably close to each other. Remember there is spacing in altitude between the different types of planes. You must be most careful of the three flights of Kates. If you place them too close together they may collide with each other before reaching the target. However, in this case one flight of Kates will not appear with the others. Place the icon for the 'Otani' flight anywhere in line with the 'Abe' flight. See the illustration for how I set up my flight. I have the flights of Kates set about 1000 feet apart (two tenths of a mile).

Now let's fine-tune waypoint #2 and the targets. Click on any of the plane icons then go to the 'Route' tab and click on the right arrow to get to waypoint #2. You may need to zoom out a couple of clicks so that you get all three ships in frame. You should be able to see the waypoints for all five flights of planes although they will probably be scattered around.

The four waypoint icons that look like triangles are our bombers. As you pass your cursor over each one, the unit ID for that flight will appear. Find the waypoint for the first flight of Kates (Abe). Drag the waypoint and place it directly on top of the carrier. When you have done that correctly the 'Target' on the route tab will change to **Essex CV-9**. Now set the waypoint for the 'Otani' flight of Kates directly on top of the 'Abe' flight, targeting the carrier again. Then set the waypoint for the third flight of Kates on the destroyer and for the Vals on the cruiser. If you are having trouble getting the target to specify the ship, zoom in closer so that you can move the waypoint more accurately. After you have targeted all three ships, set the Zero's waypoint somewhere in the middle.



Good job! We will leave waypoint #3 alone for the time being. Zoom out to around the 13 to 15-mile map width and center Wake Island in the map window. Following the same technique we used to plot the first attack wave we are now going to plot an attack on the airbase at Wake.

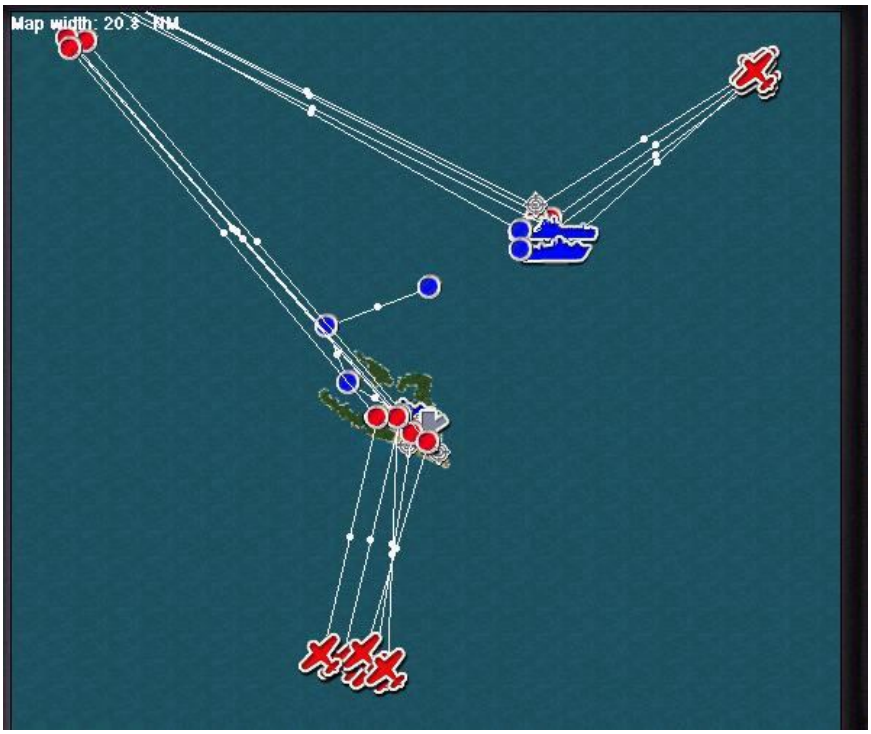
Click on 'New aircraft' and select the **G4M2 Betty**. If you have been following along, the nationality, number of planes, formation, skill, aggressiveness and goal will all match those of the other Japanese planes we selected.

You will need to set the 'Mission' as **Strike** and the 'Payload' to **Bombs Heavy**. VERY IMPORTANT – set the 'Start delay' to **00:03**. This will delay the appearance of these planes for three minutes. In other words, the game will run for three minutes before these bombers appear and begin their attack. **Note:** The 'Start delay' is computed in minutes whereas the 'Action delay' in Events is computed in seconds.

We will have this wave of bombers attack Wake from the South. Place the first waypoint about 5 miles South of the island. Place waypoint #2 on top of the island and waypoint #3 off toward the Northwest as we did with the first wave. When you have plotted this first flight of Bettys, click on 'New aircraft' and create a second set of Betty bombers. Use all of the same parameters and don't forget to set the start delay to **00:03**. Set the waypoints for this flight the same as you did the first set of Bettys.

After both sets of Bettys are plotted, click on 'New aircraft' and create a flight of Vals. Set the mission as **Anti-ship** and the payload as **Bombs Heavy**. Follow the same flight path as the Bettys then set up two flights of Kates. Set the mission to **Anti-ship** and the payload as **Torpedo**. Set the same flight path as before. Be sure to set the delay for both the Vals and Kates to **00:03**. For one flight of Kates, select the Unit ID **Suzuki**. Remember it, you will need it later. Finally create a flight of Zeros to escort this group of bombers. Their mission will be **Intercept**, their payload will be **Guns only**, and their delay will be 3 minutes.

You should now have six flights approaching Wake Island from the South and leaving to the Northwest. See the illustration for how your mission should look up to this point.



Rough plot of second attack

Just as with the first flight we need to fine-tune the waypoints for this attack. Zoom in on the cluster of planes at waypoint number one.

Using the same technique that we used with the first flight, position the second wave of bombers. Set the altitude of the Bettys to **2500** feet, the altitude of the Vals at **1500** feet, the Kates at **500** feet and the Zeros at **2000** feet. Reduce the speed of the Zeros and the Vals to **160** knots. The speed of the other planes will be fine as is.

Set the icon for the 'Suzuki' flight with the other flight of Kates. After you have all six sets of planes positioned properly, highlight any one of the planes and go to waypoint #2 on the route tab.

Set your second waypoints and targets as follows. For the first group of Bettys set the 'Action' as **Attack w/bombs** and the target as any one of the B25's. Remember to move the waypoint around until the target shows **B-25D Mitchell**. Set the altitude to **2500** feet. For the second set of Bettys the action will again be **Attack w/bombs** and the target will be **USS Tanker Ship**. Set the altitude to **2500** feet. For the Vals the action is **Attack w/bombs** and the target is **DD-722** (the destroyer). Set the altitude to **1500** feet. For both flights of Kates the action is **Attack 'w/torpedoes**, the target is again the **DD-722** and the altitude is **500** feet. Finally place the Zero's waypoint #2 anywhere over the island, set the action to **Intercept all** and the altitude to **1000** feet. Your second wave attack area should look something like the illustration.

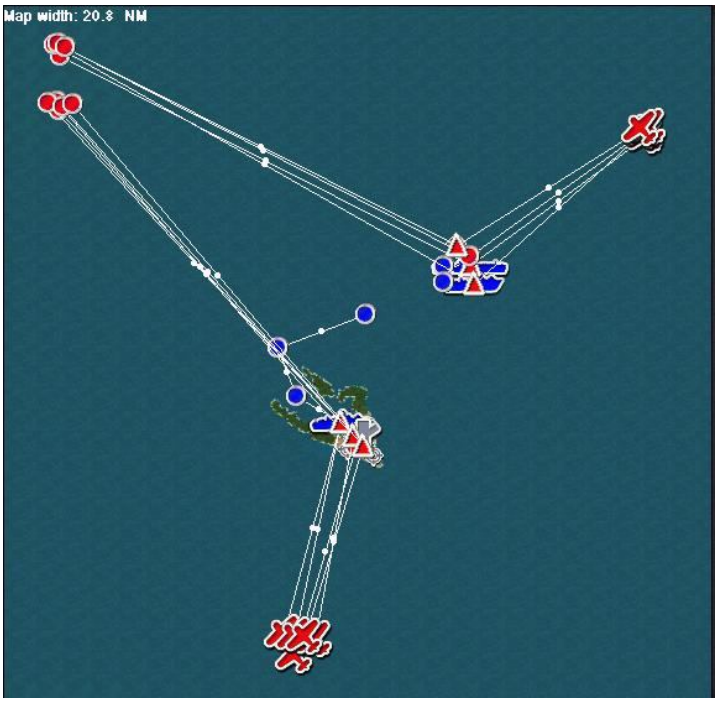
Finish off the second wave by highlighting any plane and clicking on waypoint #3 on the route tab. Set the altitude for all six flights to **3000** feet at waypoint #3.

Zoom back out to around the 20-mile width and take a look. As you can see, there is a lot happening in our mission area and we haven't even begun to plot any of the defender's planes yet, including our own.

When you start to write more complex missions with lots of planes, ships and infrastructure the map begins to get cluttered. To make it easier to see what you are doing you can apply filters to remove some of the clutter. Click the **Chart filters** button along the bottom of the map. It is the button that looks like two overlapping circles with dots in them. That will bring up the window shown here.

Each item shown with an 'X' is a type of item in use in our mission. The shaded boxes are things that we are not using. You will note that we are using 'Infrastructure' in this mission but not 'Front lines' for example. We can use the filter feature to remove from view those items or objects that are cluttering up the map. This allows us to see more clearly what we are doing without having a whole lot of 'extras' around. To filter out an item click the 'X' to remove it. You can also selectively remove one nationality's items rather then all of the items of that type. Click on the **Aircraft**, **Infrastructure** and **Airfields**

boxes then click **OK**. That will remove all of the planes we have added so far and the infrastructure around the base along with the cruiser and tanker ships which, you'll recall, were added as infrastructure and not moving ships, and the big gray 'X' marking the airfield. Your new map should look like the illustration below.



Map before filtering

Map after filtering

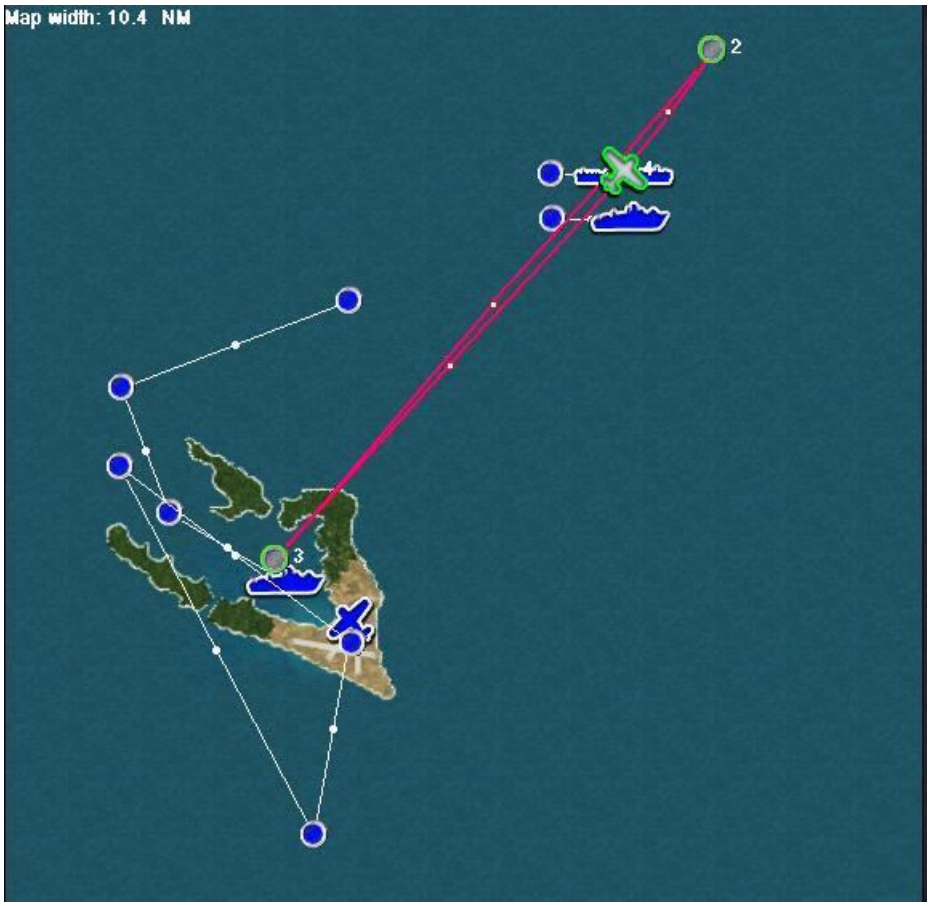
Isn't that better? You didn't 'Delete' anything. You are merely hiding some of the objects that we added in order to make it easier to work on the mission without the clutter. Now let's set up our defenders.

Plotting the Defenders' Waypoints

Zoom in one click and using the 'Pan' feature; adjust the map so that both the island and the fleet are visible. Click on **New aircraft**. Select the **F4U1-A Corsair** as the plane and the remaining parameters as follows: **USA**, **Intercept**, **Argo**, **Guns only**, **3**, **Fingertip**, **Veteran**, **High**, **None**. Set the 'Start delay' at **00:04**. We will have this flight of defenders taking off one minute after the second wave of bombers has appeared in the game. Place the first waypoint on top of the runway. Drag the second waypoint a little South of the island. Place the third waypoint near the mouth of the lagoon and the fourth waypoint back over the runway. Now, click on waypoint number #1 (the plane icon) and go to the route tab. Set the 'Action' to **Take off**. When you do that, 'Wake' should appear as the 'Base'. Set the 'Action' at waypoint #2 to **Intercept all**. The default parameters for the remaining waypoints should be good enough to keep this flight in the combat zone.

Zoom out a couple of clicks until your plane comes into view. Drag your plane over to the fleet and place it on top of the carrier. Zoom back in to the level you were at before. For your plane select the following parameters: **F4U1-A Corsair**, **USA**, **Fighter sweep**, **Agate**, **Guns only**, **3**, **Fingertip**, **Veteran**, **High**. Remember that this is the skill and aggressiveness level of your wingmen. You determine your own skill levels.

Right click on your plane icon and a pop-up window will appear. Click on 'Insert' and create waypoint #2. Drag waypoint #2 in the direction of the incoming first wave, about a mile and a half from the carrier. Insert a third waypoint and place it over the lagoon at Wake Island, then insert a fourth waypoint and place it back over the carrier. Click on your plane icon and go to the Route Tab. For 'Action' select **Take off**. When you do so 'Carrier: "CV-9" should appear in the 'Base' window. The default speeds and altitudes for the remaining waypoints will be fine. The American waypoints should look something like the illustration below.



That completes the meat of the mission. We still need to set our triggers and events which will control some of the action and the ending of the mission but, we are far enough along that you can test fly the mission and look for bugs.

Click the **FLY NOW!** button. This brings up a screen allowing you to save your mission. Title the mission **A "Wake" Up Call**. This is the name that will appear in the Missions menu in the game. Make the file name anything you'd like. This is the name the file is stored under in the MISSIONS folder of CFS2. After you have saved the mission go flying.

You are testing to insure that the Japanese planes spawn and approach their targets properly and that your wingmen and the defenders on the island do the same. As we discussed in Part 2, you may want to set your plane to 'invincible' to avoid being shot down during the testing phase. There is no trigger to end the mission, so after you have flown for a while and are satisfied that everything is OK or you've identified areas that need to be adjusted, press **ESC** and end the mission.

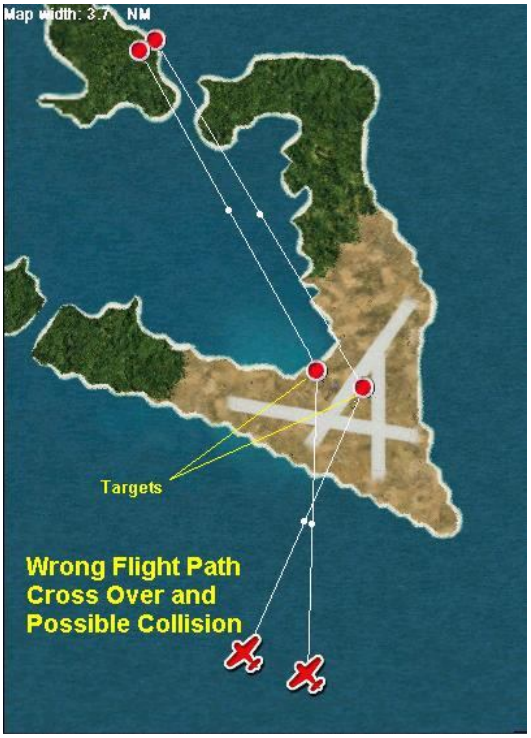
If you've followed the instructions carefully up to this point the mission should have run smoothly. You will notice that the first wave of planes approaches the carrier almost simultaneously. The second wave is a little more spread out as they attack the island. That is fine as the spacing is more challenging and keeps you and your wingmen busy. You may need to move the two flights of Japanese planes either closer to, or farther from their targets until you get the timing just right. Remember that you can measure distance in the game by left clicking on an open area and dragging your cursor. A yellow line will appear and the length of the line will be displayed in the left-hand panel near the bottom. Once you are satisfied with the flow of the mission we'll move on.

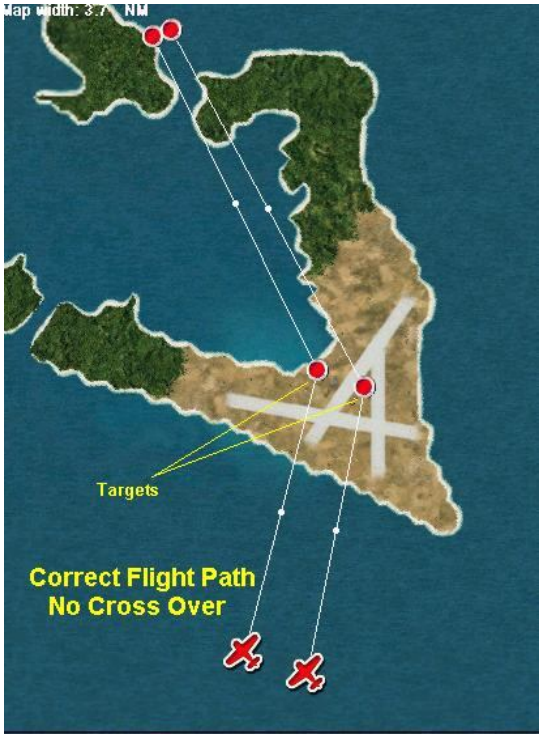
If you had a problem with the flights of bombers colliding with one another you will need to make some adjustments. You shouldn't have problems with different kinds of planes colliding because we varied the altitudes of the different models. If one flight of the same type of plane flies into each

other, for example if the Vals keep colliding, this is a problem within the game and can't be easily fixed when selecting multiple numbers of planes in the same flight.

One solution to that problem is to plot three individual Val dive-bombers instead of one group of three planes. I have had luck when doing this. You need to carefully plot the routes so that there is spacing between the planes, but I have found this an effective way to avoid the collisions that sometimes occur with multi-plane flights. It is more time consuming however. On the other hand, it does offer another advantage over the multi-plane flight. You can have each plane attack a different target. I have also found that when doing this I get better bombing accuracy than when having an entire flight aim for one target such as we are doing in this mission.

Another cause of mid-air collisions occurs when you have your flight paths crossing one another. If your two flights of Bettys attacking Wake are colliding this could be the problem. As you look down on the mission map we have one flight of Bettys striking the B-25's and the other flight striking the





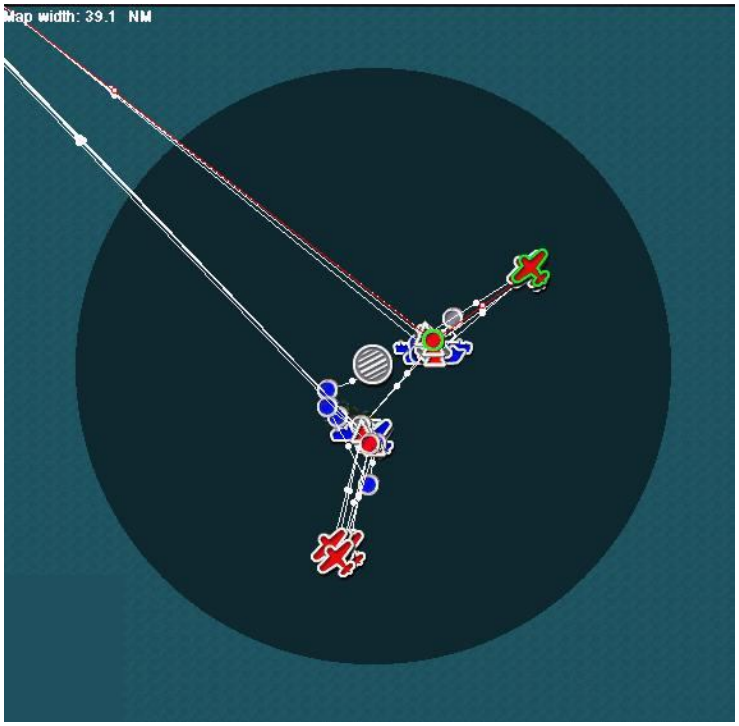
Tanker ship. If the flight on the right goes after the tanker and the flight on the left attacks the B-25's they must crisscross in route and that can cause a collision. Keep this in mind when plotting your waypoints and have each flight stay in its own 'lane' so to speak. See the illustrations above.

Setting the Event Area

We are going to use an Event Area as one of the ways to end the mission. The technique we are going to use is the same as that used in the "Aces" missions that come with the game. Zoom out to about the 39-mile width and click on the **New event area** button on the upper toolbar. Place your cursor midway between the carrier and Wake Island and while holding down the left mouse button, drag the cursor downward forming an event area with a radius of **12** miles. Release the mouse button and move to the information panel on the left side of the screen. Name this event area **Inner Zone** and set the 'Radius' to **12.00** NT (nautical miles), then click **Apply**.

We will now place a second Event Area directly on top of the first one. Click the **New event area** button and place your cursor right in the center of the first event area icon. Drag this new event area out to a radius of **16** miles. If you are off center, you can click the 'Undo' button along the bottom and try again. You can also zoom in to exactly place the second event area on top of the first. When you are finished, name this new area **Outer Zone**; set the 'Radius' to **16.00** NT and click **Apply**.

Finally, zoom out one more click. Click on the **Chart Filters** button and click the box next to 'Infrastructure' and 'Airbases' to restore all of the hidden icons. Then drag the 'Waypoint #3' markers for each flight of bombers to a point well away from the combat area. One of my pet peeves in CFS2 is the behavior of the bombers after they have dropped their bombs. Historically, when a flight of bombers finished their bomb run they headed for home and let the fighter escorts deal with the attacking defenders. You may have noticed that in CFS2 some of the bombers break formation and behave like fighters themselves. There is a cure for this but it is beyond the scope of this Handbook. I believe that it helps to set your final waypoints well away from the combat area to help prevent this. It won't stop it, but it does 'encourage' the bombers to keep moving. If you set your final waypoint too close to the action, when the enemy planes get there they will begin milling around looking for a place to land. It is more realistic to set that last waypoint well away from where the action is taking place when dealing with bombing missions.



Completed Combat Area

Completed Mission Map

Creating the Triggers and Events

We will finish off our mission by using triggers and events to control some of the action and create several different options to end the mission. You will recall that the goal for both our carrier and the destroyer in the lagoon is 'Must survive'. We will start by creating triggers and events that will penalize us if they are sunk. Begin by clicking on the **Triggers** button and then on the **New** button.

Title this first trigger **Carrier sunk**. The Object 'Category' will be **Moving object**. Scroll down and select the **Essex CV-9 USS Essex class Aircraft Carrier**. Your remaining parameters will be:

'Type' -	'Status'
'Param' -	'Object health'
'Operator' -	'Equals'
'Value' -	'0'

Click **OK**, **Close**, then click on the **New event** button. Title this event **Respawn destroyer attack**. Select the **Carrier sunk** as the trigger. Under 'Action type' select **Spawning** then click on **Spawn unit**. Scroll down and select the Unit ID **Suzuki**. It is very important that you choose the correct flight of bombers. Suzuki is the name of the second flight of Kates attacking the destroyer. See the illustrations at left and below. Close out the event.

Open up a new trigger. Title this trigger **Destroyer sunk**. The 'Category' will again be **Moving object** but this time select the destroyer in the lagoon. You have two destroyers in the game, be sure you pick the correct one. If you used the defaults it should be 'DD722'. Your remaining parameters will be **Status**, **Object health**, **Equals**, **0**. Click **OK**, **Close**, and then open a new event.

Title this event **Respawn Carrier attack**. Select the **Destroyer sunk** trigger. Under 'Action type' select **Spawning** then click on **Spawn unit**. Scroll down and select the Unit ID **Otani**. Again, it is very important that you choose the correct flight of bombers. Now, select **Action delay** under **Action type** and set a delay of **30** seconds. Then select **Play priority sound**, scroll down and select sound

number **w639** and enter it. You can click the 'Test' button if you want to listen to the sound first. In the event that a second wave of torpedo bombers is launched against the carrier we will get a radio message ordering us back to defend the ship.

Picture what we have done here. These two triggers/events are mirror images of one another. Our mission is to save both the carrier and the destroyer in the lagoon. If either one of those ships is sunk the game will launch a second attack against the **other** ship. This element puts a little extra pressure on us to succeed and increases the challenge of the mission. The two additional flights of Kates, Otani and Suzuki, will not appear unless their respective spawning event is triggered. That is why we can place them on exactly the same flight path as the other bombers. They won't be there at the same time so we can't have a collision.

Open another trigger. Title this one **Elapsed time**. The 'Category' will be **General** and the parameters are:

'Type' - 'Environment'
'Param' - 'Time elapsed'
'Operator' - 'Greater than or equals'
'Value' - '00:04'

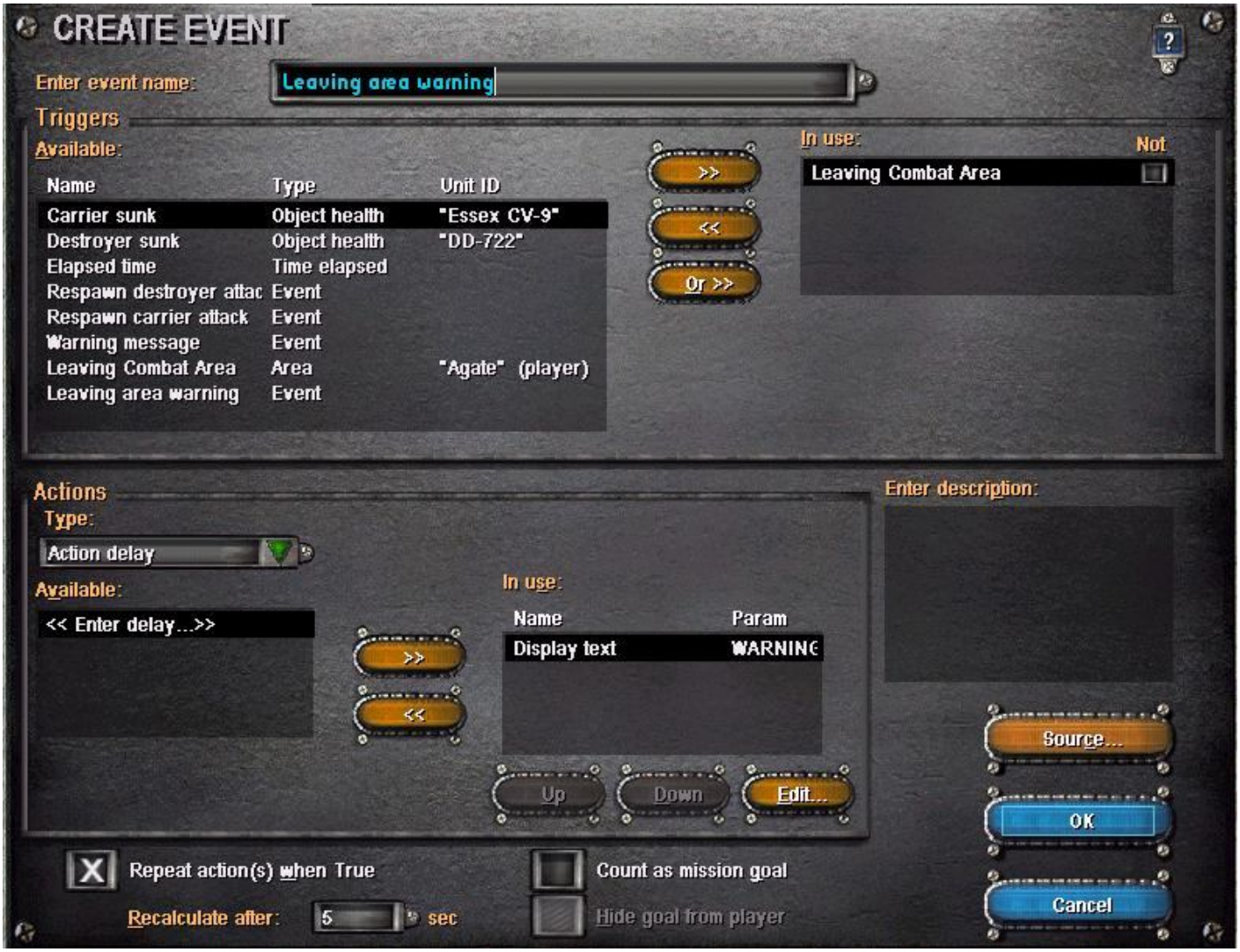
Close it and open a new event. Title this event **Warning message**. Choose the trigger you just created and select **Display text** as the action. Type in this message: **Radar reports bombers approaching Wake Island from the South**. Click **OK** to activate the text then close out the event. We now will get a warning message on our screen when the Japanese bombers are nearing Wake. It will serve as a reminder in case we are too preoccupied with the wave attacking the carrier.

There will be three ways that we can end this mission (Four if you count getting yourself killed!). The mission will end if we shoot down 10 or more enemy planes, if we leave the combat area, or if we are shot down and successfully crash-land or ditch our plane in the ocean. We need to create triggers and events to control each of these possibilities.

Open a new trigger and name it **Leaving Combat Area**. Your 'Category' will be **Aircraft**, then choose your plane **Agate" F4U1-A Corsair (player)**. Set these parameters:

'Type' - 'Status'
'Param' - 'Area'
'Operator' - 'Outside'
'Value' - 'Inner zone'

Close it and open a new event. Name this event **Leaving area warning**. Choose the trigger you just made then select **Display text** under **Action** and click the **>>** arrows. Type in the following text**: WARNING! You are leaving the Combat Area**. Click the **Repeat action(s) when True** box and set the **Recalculate after** time to **5** seconds. Close out the event. This will give us a warning message should we stray too far away from the battle area. We can still return to the action without ending the mission. See the illustration on the next page for this completed event.



Open a new trigger. Title this trigger **Left combat area**. Choose your aircraft as the object and set the parameters to:

'Type' -	'Status'
'Param' -	'Area'
'Operator' -	'Leaves'
'Value' -	'Outer zone'

Save and open a new event. Call this event **End mission – Left area**. Select the trigger you just created. Under **Action** select **Display text** and type in the following: **You have left the combat area. This mission is being terminated.** Click **OK**, then select **Action delay** and set a delay of **8** seconds. Select **Miscellaneous** then **End mission**. Click **OK** and close out this event.

Next we will create the trigger/event combination to end the mission when we have shot down 10 enemy planes. Open a new trigger. Call this trigger **Player shoots 10**. Select your aircraft as the object and set these parameters:

'Type' -	'Status'
'Param' -	'Aircraft kills'
'Operator' -	'Greater than or equals'
'Value' -	'10'

Save it and open a new event. Title this event **End mission – kills**. Select the new trigger. Under actions select **Play priority sound**, scroll down and select sound number **w662** and enter it. Then select **Display text** and type in the following: **Great job! You have stopped the enemy attack. The mission will now end.** Select **Action delay** and set a delay of **10** seconds then select **Miscellaneous** and **End mission**. Finally, select **Debrief text** and type in the following: **Excellent work! You have joined the select group of Aces flying in the Pacific Theater.** Now, if you're lucky enough to shoot down 10 enemy planes you will get a verbal pat on the back from your wingmen and recognition at the debriefing.

☛ As you become more proficient, you may find it easier and quicker to create all of your triggers first and then create all of your events, rather than switching back and forth as we do here. Chose the method that works best for you. This chapter is aimed at the novice mission builder and by doing everything in sequential order it is easier to see the relationship between each trigger and its related event.

The last event ending the mission will be more complicated. We are going to create a trigger/event combination that will end the mission if we are shot down but manage to ditch our plane safely. If you've played CFS2 much, you know that one of the problems with the game is that no matter how well you do otherwise, if you are forced to ditch you fail the mission. You either sink in the ocean and die or must press ESC and abort the mission. Either way you lose. We can create a trigger/event combination that will allow us to safely ditch and end the mission without dying or aborting. Once you master this technique, you can add it to any mission. There are several different ways to do this depending on what you want to use as triggers. This is just one method.

First we must define what a successful ditch is. There are three elements, which must be true for us to safely ditch our plane:

1. we must be alive
2. the plane must be stopped
3. the plane must be on the ground or on the surface of the ocean (zero altitude)

All three of these elements must be true for us to have safely ditched. When all three of these elements occur simultaneously, we will declare our plane ditched and the mission will end. If you use only two of the elements instead of all three, there are circumstances, which could occur that would end the mission before you actually ditched the plane.

You could include elements such as requiring that the landing gear be retracted. That would make sense if you ditched at sea but with Wake Island so close to the combat area it's possible you could crash land there with your gear down. We could also require that the flaps are down, but what if your controls are damaged and they can't be lowered? Use the three elements we've identified here and the trigger/event will work correctly.

If you look carefully at our three elements however, you will notice a flaw. All three of these statements are true when the mission first begins and you're sitting on the flight deck of the carrier getting ready to start your engine. Obviously we don't want our mission to end before it begins, so we will add one more trigger. We will delay this event for one minute. In other words, the mission must run one minute before you can safely ditch and end the mission. That's only fair. If you crash land in the first minute of the mission, you deserve to flunk! Now, let's create the triggers and event.

Open a new trigger and call it **Ditching delay**. Your category is **General** and the parameters are:

'Type' -	'Environment'
'Param' -	'Time elapsed'
'Operator' -	'Greater than'
'Value' -	'00:01'

Save it and open another new trigger. Call this trigger **Player landed**. Select your aircraft as the object and set the parameters as:

'Type' -	'Flight'
'Param' -	'Altitude (AGL)'
'Operator' -	'Less than or equals'
'Value' -	'0'

Close it and open another new trigger. Call this one **Player stopped**. Again select your aircraft and set the parameters as:

'Type' -	'Status'
'Param' -	'Speed'
'Operator' -	'Less than or equals'
'Value' -	'0'

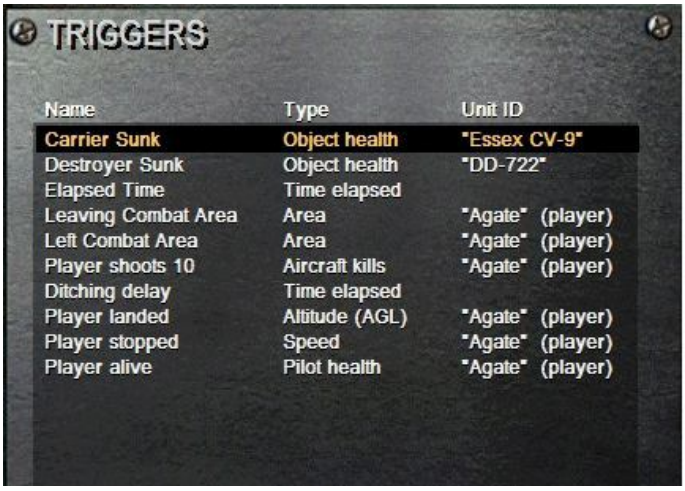
Close it and open one more trigger. Title this one as **Player alive**. Again, select your aircraft and set these parameters:

'Type' -	'Status'
'Param' -	'Pilot health'
'Operator' -	'Greater than'
'Value' -	'0'

Close it out and open a new event. Title this event **End mission – ditch**. Begin by selecting all four triggers we just created, **Ditching delay**, **Player landed**, **Player stopped**, and **Player alive**. After selecting all four triggers drop down and select **Display text** as the action. Type in: **You have safely ditched your plane. This mission will now end.** Select **Action delay** and set a delay of **8** seconds, then select **Miscellaneous** and **End mission**. Finally select **Debrief text** as the Action and type in this text: **You survived to fight again but you lost your plane. Next time, stay alert and watch your six!**

Click 'OK' and close out the event window. That completes the action for this mission. We have created 10 triggers and 7 events. See the illustrations on the next page for the complete list.

☛ Some aircraft may show an AGL altitude of greater than zero even when sitting on the ground. If you are designing a ditching scenario for an aircraft with this problem simple change your trigger parameter for "Player landed" to something like 'AGL' 'Less than' '3' feet, or whatever number will place you below the stationary AGL of the plane.



Full list of triggers

Full list of events

Mission Setup

Click on the **Mission setup** button. Under **Briefing Overview** type in the following: **The Japanese have launched an early morning raid against the airbase at Wake and the ships offshore. Get airborne and intercept the attacking bombers. The mission will end when you have destroyed 10 enemy planes or you have left the combat area.**

Under **Background** type in: **The Japanese have caught us off guard. They will either focus their attack on our carrier or on trying to sink the destroyer in the lagoon and blocking the entrance to the harbor. They must not succeed!**

For your **Intelligence** information enter: **The Japanese are well armed in this sector. Expect a wide variety of aircraft making up this assault.**

Remember, you're writing these briefing notes as though someone else, who knows nothing about the mission, is going to read them. Try to make them informative as well as entertaining. The briefing should always tell the player what he needs to do to successfully complete the mission, or in cases like our shoot-em-up, how the mission will be terminated.

This is a fun mission and isn't even remotely historically accurate so you can choose any date you wish. However, set the time to **07:00**, as this is an early morning attack. If you want some clouds or rain go ahead and add it. One caution, clouds make for slower frame rates and we already have quite a few planes flying around. You might want to leave the skies clear to keep performance up.

Your mission is now complete. Save it and let's go flying. Now that all of the triggers and events are in place you really need to test each one of the endings for the mission. You will need to fly it several times to make sure that everything is working correctly. When you're finished, meet me back here.

Tips, Thoughts and Troubleshooting

A "Wake" Up Call is the most complex of the missions we have written so far, but it isn't difficult. We merely added some additional features and more complex triggers and events. By now you should feel comfortable with Mission Builder and be getting a good handle on the various tools that give you so many options in designing your missions. Here are some thoughts and tips regarding *A "Wake" Up Call.*

You should have tested the flight patterns and distances earlier so this time was mainly to check the triggers and events. You don't have to get shot down to check the ditching function. After you've flown around for a minute simply cut your throttle and put her down on the ocean. By the same token, you don't have to shoot down 10 planes to test this trigger. For testing you can modify the number of kills in the trigger to 3 prior to flying; if the trigger works at 3 kills it will work at 10. Just remember to modify it back up to 10 when you're finished. To test leaving the combat area, just take off and keep heading west. When you get the warning message, turn and reenter the area. The warning message should stop. Once it does, turn west again and fly until the mission ends.

You should also have tested the spawning of the additional torpedo bombers. If you don't take any hostile action toward the first flights of Kates, the odds are that they will sink one or both of the ships. If you've set your triggers and events properly the additional bombers will spawn and begin their run. In fact, if you've followed the steps in the lesson carefully up to this point, everything should work flawlessly.

If your carrier gets sunk first and the destroyer in the lagoon gets sunk later the game will still spawn another flight of Kates to attack the carrier even though it isn't there. Consider them some extra targets. If that's a problem for you, here's a simple solution. Open the Event window, highlight **Respawn carrier attack** and click **Edit**. Highlight the trigger **Carrier sunk** and click the **>>** arrows placing it in the 'In use' window. Now click the **Not** box to the right of the trigger name. This will place an '**X**' in that box. Click **OK** and close out the event.

An event occurs when all of the triggers for that event are true. The second attack on the carrier is spawned when the destroyer is sunk – true. By adding a trigger and clicking the 'Not' box we are saying that this trigger must be false for the event to occur. Now the second carrier attack will be spawned when the destroyer is sunk – true, and the carrier is sunk – false. In plain English, the second wave spawns when the destroyer is sunk but the carrier is not.

Here are some additional adjustments you can make:

Mission Too Hard:

- reduce the number of planes in each flight or eliminate some flights altogether
- reduce the skill level of the Japanese pilots to 'Rookie'
- increase the number of defenders

Mission Too Easy:

- add more Zeros – spawn flights of 6 instead of flights of 3
- raise the skill level of the Japanese pilots to 'Ace'
- reduce the amount of time before the second wave attacks Wake

Troubleshooting *A "Wake" Up Call*

Without any warps, there should be little to troubleshoot. The mission, while busy, is pretty clean. If the enemy planes are showing up too early over Wake you may have failed to set one or more flights to a 3-minute delay. If the bombers are running into each other, review the steps we covered on pages 51 and 52 for preventing collisions. If the ships are colliding, double check that you set them

as ‘Do not evade’. If that setting is correct, try moving them farther apart or staggering them instead of having them sailing abreast.

About the only other potential problem area is with terminating the mission. If one of the events is failing to end the mission go back over the steps we outlined above. If you follow the steps exactly, the game will end the mission according to plan.

Summary

You now have three missions under your belt and if you’ve followed the lessons carefully you should feel pretty comfortable with designing your own missions. If you quit right now and never learned another thing about the Mission Builder you would still be able to write exciting and fun missions. But in reality, we have only scratched the surface of what Mission Builder is capable of. In the next part we will explore how to add variety to your missions and make even simple missions come alive.

☛ Within Mission Builder you can use basic Microsoft editing functions such as **Copy** and **Paste**. This can speed the process of entering text in events or in ensuring that your briefings are grammatically correct.

Try typing your briefings in Word, run spell check to make sure they are correct, and then copy the text to the clipboard. Open Mission Builder (or have it running in the background and use Alt+Tab to switch back and forth) and simply paste the text into the appropriate area of the Mission Setup screen.

Another handy way to use copy and paste is when adding text messages in events. For example, say you wanted to repeat an on-screen message three times as part of an event. Instead of retyping the text into the “Display text” window each time, you can type it in once; select it by clicking “Edit” drag your cursor from one end to the other to select the text, then press Ctrl+C to copy the text to the clipboard. Select “Display text” again and when the “Text Message” window opens simply press Ctrl+V to paste the text into the box.

If the text is too long to highlight by dragging the cursor, simply click “Edit” that will open the text window with the text selected. Press Ctrl+C and follow the instructions as laid out above.

Part Two

enhanced Mission Building I

A series of more advanced techniques designed to help you get the most out of your mission building experience.

Chapter five – playing the percentages

Preflight

Even a well crafted mission can become boring after a while when you always know what is going to happen next. Fly almost any mission more than a couple of times and you will know when the enemy will appear and in what strength. Although it's still fun to fly, you lose the element of surprise and a lot of the challenge when you know what to expect. In this chapter you will learn how to keep your missions fresh and challenging by designing random chance and percentages into your triggers and events and by hiding enemy objects from the player during the briefing process.

Incorporating random chance into your mission

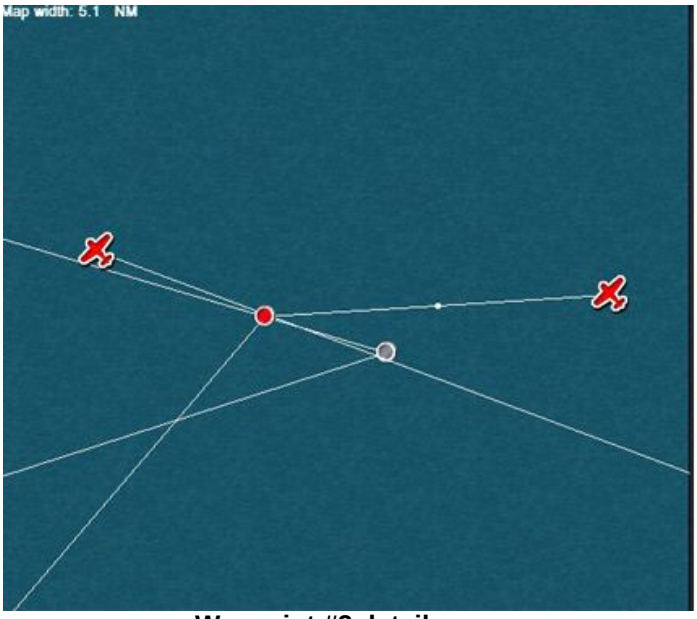
Let's begin by opening up *Beginner's Luck*, the first mission we wrote. While this first mission may have been a challenge for you when you were starting out, by now, even if you have only completed the other two missions in this Handbook, you can see that it is a very simple mission that plays out the same every time. Let's doctor it up some to make it more exciting.

Original Beginner's Luck Layout

Start by adding a wingman to your flight. Click on any of your waypoints and increase the number of aircraft in your flight to **2.** Set the formation as **Fingertip** and make your wingman a **Veteran** with aggression as **high.** Then zoom in on your waypoint #2 until you get to the 5 mile width.

Add a second flight of Zeros. You can copy the existing flight using the CTRL key and dragging as described in one of the sidebars above or you can click on the 'New Aircraft' button and add the flight that way. Set this flight's parameters as: '**A6M5 ZERO**', '**Japan**', '**Fighter Sweep**', '**Aoshima**', '**Guns Only', '3', 'Shotai**', '**Rookie**', '**Average**'. Note that this flight has three planes while the original flight of Zeros had only two. Set the goal for both flights as '**None**'.

Set the first waypoint for this flight about a mile and a half east of your waypoint #2. Set the second waypoint mid way between your waypoint #2 and the original Zero flight's icon. Set the third waypoint well southwest, just touching the north edge of Guadalcanal Island and the fourth waypoint south of the island. You will need to zoom out to plot these last two waypoints. Set the altitude at all four waypoints to **3500** feet and raise the speed to **240** knots at each waypoint. See the illustrations.



Waypoint #2 detail

Second flight plotted

Open the triggers window and create a new trigger titled **'Random 50'.** Select **'General'** as the category and use these parameters:

'Type' - 'Percentage'
'Param' - 'Random number'
'Operator' - 'Equals'
'Value' - '50'

You should have noticed that when you selected 'General' as the category the defaults matched what we wanted entered into this trigger. The 'Value' number is the percentage of time this trigger will be true. A value of 50 means that 50% (or half) of the time this trigger will be true, i.e. it will activate whatever action it is linked to. The lower the value the lower the chances are that this trigger will activate. The higher the number the more likely the chances are that this trigger will come into play. You can set the value anywhere from 0 to 100. However, it doesn't make sense to use either zero or 100 as a zero means the trigger will never activate and a 100 means it will always activate and in either of those cases you don't need the trigger to begin with. Realistically, the practical range of numbers is from 1 to 99. A trigger set at 1% will very rarely be true while one set at 99% will almost always be true.

You need to be thinking ahead when designing triggers using random percentages. You are going to link this trigger to an event. If you want the event to occur more often than not, set the value above 50%. If you want a smaller likelihood of the event occurring set the value below 50%. In this case there is an equal chance of the event occurring that we will link this trigger to.

Close this trigger then create another new trigger titled '**Aoshima destroyed**'. Select the Aoshima Zeros as the object then set these parameters:

'Type' -	'Status'
'Param' -	'Object health'
'Operator' -	'Equals'
'Value' -	'0'

Save this trigger and then highlight the trigger named 'Enemy Planes Destroyed'. Click the '**Edit**' button then rename this trigger to '**Honda Destroyed**'. Close out the triggers window and open a new event.

Highlight the event named '**Spawn Enemy Planes**'. Click the '**Edit**' button. First, change the name of this event to '**Spawn Honda**', then select the **Random 50** trigger we just created and add it to the column of in use triggers. Save this event and open a new event.

Title this event '**Spawn Aoshima**'. Select these two triggers, '**Player arrives at WP #2**' and the event '**Spawn Honda**'. Click in the '**Not**' box next to the trigger '**Spawn Honda**' so that an "**X**" appears there. For your actions select '**Miscellaneous'** '**Disable warp**' followed by '**Spawning'** and spawn the unit **Aoshima**. Save this event.

Picture what we have done here. First we created a trigger with a random chance of 50%. Think of it like a coin toss; half of the time one thing will occur and half of the time something else will occur. Then we linked that trigger to the spawning of Honda. When we arrive at waypoint #2 there is now a 50-50 chance that Honda will spawn just like it did in the original mission. Next we created an event that will spawn Aoshima if we arrive at waypoint #2 and Honda **doesn't** spawn. Either way when we get to waypoint #2 we will encounter enemy fighters. Half of the time it will be a flight of two from our front and half of the time it will be a flight of three rapidly approaching from our rear. Already *Beginner's Luck* is looking more interesting!

☛ Most likely you will be writing missions from scratch, but occasionally you may find yourself editing an existing mission and adding new triggers and events as we are doing here. In some cases it might be easier to delete all or some of the existing triggers and events first, rather than trying to make the modifications and get everything to work right.

You can delete triggers and events by opening either window, highlighting each one, and clicking the **Delete** key. You must delete the events first and then the triggers. This method allows you to leave selected triggers or events in place.

A faster way to delete all of the triggers and events at once is to open your **Missions** folder in CFS2 and locate the mission files there. You will see two files for each mission that has triggers and events. One has an extension of **.mis** and the other has an extension of **.dyn**. Delete the file with the **.dyn** extension and all of the triggers and events are erased.

CC_001.mis	this file contains the basic mission data, planes, ID's, waypoints, etc.
CC_001.dyn	this file contains all of the triggers and events

By adjusting the percentages you can control which flight is more likely to attack. You should also be able to see how you can apply this same technique to any spawning event, friend or foe, and change the dynamics of the mission each time. OK, let's adjust our warping events to cover these changes.

Our warping event connected to Honda is still good, but let's change the name from "Enable Warp 1" to **Enable Warp "Honda".** Then create a new event named **Enable Warp "Aoshima"**. For your triggers use "**Spawn Aoshima**" and "**Aoshima destroyed**". Set a **5** second delay then enable warp. Now, regardless of which flight of Zeros is spawned we can warp away after shooting them down.

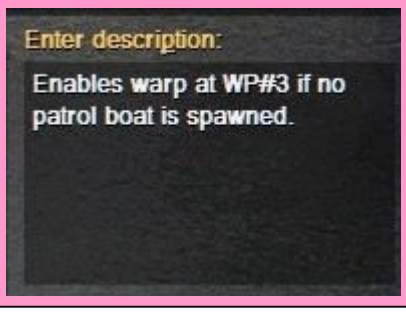
Create two new triggers. Name one "**Random 30**" and the other "**Random 60**". Set these triggers up just like you did the Random 50 trigger you created earlier except use the values **30** and **60** respectively. Your category is "**General**" and your parameters are:

'Type' -	'Percentage'	Type' -	'Percentage'
'Param' -	'Random number'	'Param' -	'Random number'
'Operator' -	'Equals'	'Operator' -	'Equals'
'Value' -	'**30**'	'Value' -	'**60**'

Then change the name of the trigger "Enemy Ship Destroyed" to "**First Patrol Boat Destroyed**". Close out the triggers and open the events. Select the event "Spawn enemy ship" and click "**Edit**". Change the name of this event to "**Spawn first patrol boat**" and add the trigger "**Random 30**" to the "In Use" column. Click "**OK**" to save the event. Now when we arrive at waypoint #3 there is a 30% chance that the patrol boat will spawn. Unlike the action at waypoint #2 where we set it up so that we would encounter one of two flights of Zeros, here it is Either-Or. Either an enemy patrol boat is spawned or nothing happens at this waypoint.

Since warping from waypoint #3 is linked to the patrol boat being sunk we need to construct another event to allow warping in the event the boat doesn't spawn. We also need to make some changes to the existing event that allows warping. Highlight "**Enable warp 2**" and click "**Edit**". Change the name to "**Warp at WP#3 – Boat**". The triggers in use are fine but drop down to the action section and remove the debrief text. To remove it simply highlight it and click on the double left arrows **<<**. Close this event and open a new one. Name the new event "**Warp at WP#3 – No Boat**". For your triggers select "**Player arrives at WP#3**" and "**Spawn first patrol boat**". Click the "**Not**" box next to "Spawn first patrol boat". For your actions select **Miscellaneous** – "**Disable warp**" followed by a **15** second delay followed by **Miscellaneous** – "**Enable warp**". See the illustration.

☛ You can name your events anything you'd like. I am using descriptive titles to help you understand what each event does. If you prefer a simpler title but still want a description of what is happening in the event, put a clear explanation in the "Enter description" box next to the actions window.

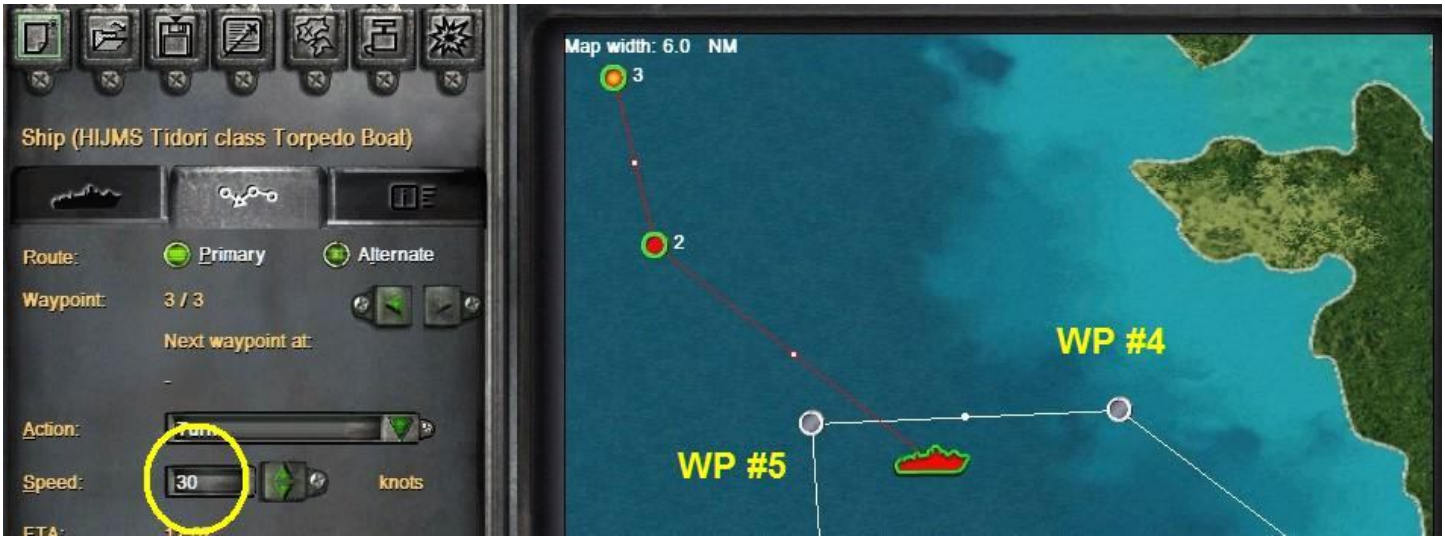


Very good, now when we reach waypoint #3 we may or may not encounter a Japanese patrol boat. If we do, we warp away after sinking it; if we don't encounter the boat we fly along on patrol for 15 seconds and then warp away. Zoom in on Waypoint #3 until you get to about the 23 mile width. Find the dot marking the mid point of your flight path between waypoint #3 and Henderson and drag it to the northwest, along the coast of Florida Island, creating a new waypoint #4. Then do that again to create a waypoint #5 a couple of miles west of your new waypoint #4.

We created two new waypoints because we want the player to come out of warp at waypoint #4 facing the open sea, not toward Henderson. When you come out of warp you are facing the next waypoint. If we didn't have waypoint #5 in there we would come out of warp headed south, back toward Guadalcanal. I wanted to simulate still being on patrol, not having already turned back to the base.

Both of these new waypoints will have the flight information from waypoint #3, 1000 feet and 219 knots, and both will be marked as "Warp to". Go to the Route Tab and uncheck the "Warp to" box for waypoint #5. We **do not** want to warp to waypoint #5. Now zoom in on waypoint #4 so we can add another patrol boat.

Click the "**New ship**" button and add another Tidori class Patrol Boat just like you did at waypoint #3. Be sure to set the nationality to Japanese. The other default settings will be fine but up the speed at all waypoints to **30** knots. You can put in as many waypoints as you'd like just be sure that the first one is fairly close to your flight path.



Create a new trigger titled "**Second Patrol Boat Destroyed**". Select the new patrol boat as your moving object. Be sure you pick the correct patrol boat. Your parameters are as follows:

'Type' -	'Status'
'Param' -	'Object health'
'Operator' -	'Equals'
'Value' -	'0'

☛ If you have a very full mission it can sometimes be difficult to pick the right object, plane, or ship when setting up your triggers. An easy way to ensure that you have the correct one is to highlight the object in Mission Builder first, then open the Triggers window. The object you highlighted will appear as the default infrastructure, ship or airplane selected in the trigger.

Save this trigger and create another one named "**Player arrives at WP#**4". Choose your aircraft and set your parameters as:

'Type' -	'Status'
'Param' -	'Waypoint Reached'
'Operator' -	'Equals'
'Value' -	'4'

Save it and open the Events window. Create a new event named "**Spawn second patrol boat**". You will select three triggers: "**Player arrives at WP#**4" followed by the event "**Spawn first patrol boat**" and finally "**Random 60**". Click the "**Not**" box next to "**Spawn first patrol boat**". Set your actions as "**Disable warp"** and "**Spawn unit**" then be sure that you select the correct patrol boat for this spawning event.

We now need two events to enable warping, one if the patrol boat spawns and one if it doesn't. These two events will be patterned along the two we set up for the first patrol boat. Your first event will be named "**Warp at WP#4 – Boat**". Your triggers are "**Spawn second patrol boat**" and "**Second Patrol Boat Destroyed**". Set a **5** second delay then enable warp. Title the second warping event "**Warp at WP#4 – No Boat**". Your triggers will be "**Player arrives at WP#**4" and the event "**Spawn second patrol boat**". Be sure to check the "**Not**" box. Then set your actions as "**Disable warp"** followed by a **15** second delay then "**Enable warp**".

Let's recap what we've done. Using random chance and percentages you now have variables built in at three waypoints. At waypoint #2 you will encounter one of two flights of Japanese Zeros, random chance will decide which one. At waypoints #3 and #4 you may or may not encounter an enemy patrol boat. Study the triggers and events and you will see that it is possible to encounter one or the other of the patrol boats but never both. At waypoint #3 there is a 30% chance of spawning a boat. At waypoint #4 there is a 60% chance of spawning a boat but <u>only</u> if no patrol boat was spawned at waypoint #3.

Testing and Troubleshooting

Now would be a good time to test the mission and make sure that everything works. When you click "**Fly Now**" you will be asked if you want to save the mission. First, rename this new mission from *Beginner's Luck* to **Luck of the Draw**. Then save it using a different file name than you used before. Remember, never overwrite an existing mission and save it using the same file name. That's a sure recipe for trouble down the road. Once it's saved, go flying.

If you followed the instructions carefully the mission should have run flawlessly. The problem is, with so many variables how do you test each one? The method I use is to test each element individually. If the individual parts work I know the whole package will work.

To test the action at waypoint #2 you can increase the percentage in Random 50 to **100**. In your test the Honda flight should always appear. Then modify the Random 50 trigger to **0**. Now Honda should never appear but Aoshima always will.

To test the action at waypoints #3 and #4 start by modifying Random 30 to **100**. Now when you reach waypoint #3 the first patrol boat will appear. After you warp to waypoint #4 you should not see the second patrol boat. Then modify Random 30 to read **0** and modify Random 60 to read **100**. This time you should not see the first patrol boat but will see the second. Finally modify both Random 30 and Random 60 to read **100**. In this test you should see the first patrol boat but not the second. Even though the percentage at waypoint #4 is set to 100 the second boat should not appear because the first one did. If something isn't working correctly go back over the steps in this chapter carefully to isolate the error.

Once you are satisfied that everything is working correctly **BE SURE TO RETURN ALL OF THE PERCENTAGES TO THEIR CORRECT NUMBERS**. Get this last step wrong and you'll be wondering why the mission plays the same over and over again.

Testing can be a lot of work but it is necessary if you want to produce consistently high quality missions. If you just assume everything is OK, sooner or later you'll have a glitch that will ruin the mission. It might not bother you much but it will be very frustrating for someone else if you are posting your missions on the Internet. Thorough testing can resolve problems and keep your missions functioning flawlessly.

There are a couple of other tricks I use when testing. During testing I'll change my first waypoint from Take Off to **Turn**. This way I start in the air and I don't waste time going through the take off process. As soon as I've tested an element of the mission I hit the escape key (Esc) and exit the mission so I can go back to editing. Also, I use a downloadable cheat that allows me to destroy any enemy planes with a single key stroke. That way I don't have to fight my way out of trouble before warping. Finally, I make my player invincible so I don't have to worry about getting shot down or crashing. As above, when the testing is complete be sure to return everything to their original settings.

You may notice that at waypoint #3, if the patrol boat spawns it will be off your left wing. You won't see it unless you are looking around. The first indication you may have of it being there is when the flak starts exploding around your plane. If this is a problem for you, move the boat where it will be easier to see. As always, the purpose of this Handbook is to teach you techniques, not to write historically accurate missions. Feel free to tweak the locations of waypoints, altitudes, speeds, etc. to get the desired effect you want.

One final word about random percentages, you can only use a random percent trigger once per mission. If you wanted to have a 50% chance of spawning the Zeros and the patrol boats you could not create a single random trigger of 50% and use it all three times. You would need to create three separate triggers. Each could have a random percentage of 50 but they would all need a different name. Once a random percent trigger is used it cannot be reused in that mission. That is not true of other triggers. For example you could create a trigger "Player arrives at WP #4" and use it in multiple events but that won't work with random percentages.

Hiding the Enemy

By using random percentages you've been able to change a very simple mission into one that is more interesting. There is another simple step that you can apply to any mission to keep it challenging.

Before you fly you can click on the "**Advanced Info**" button on the briefing screen. This will take you to a screen where, among other things, you will see a map of the mission. This is a slightly smaller version of the same map that you use in Mission Builder. A good player will always look at the Advanced Info. After all, checking the map before you fly is no different than viewing a real map at a real preflight briefing. The Advanced Info map will contain all of the aircraft, ships, infrastructure, waypoints, front lines, etc. that you will see on the Mission Builder map. That means an experienced player can look at the map and get a really good idea of what is going to happen in the mission. Take a look at the illustration:

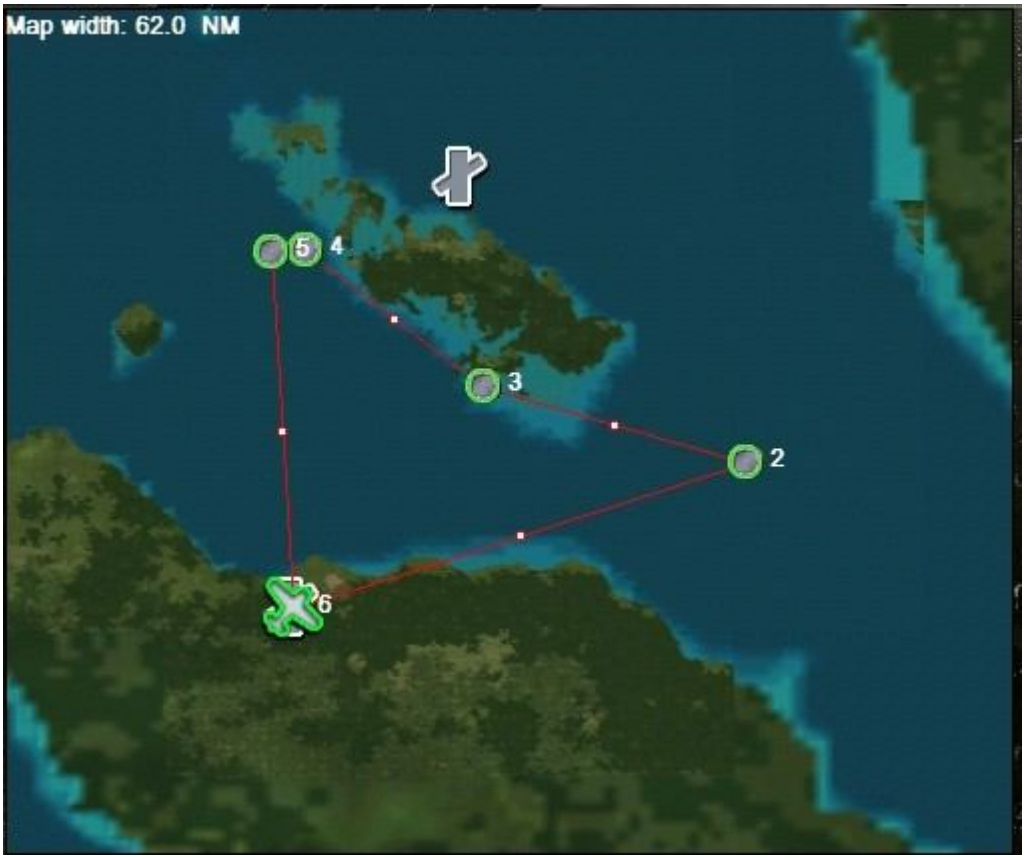
This is the map for *Luck of the Draw* that the player will see if he goes to Advanced Info. Even though the player knows nothing about our triggers and events he can see that there are two flights of enemy planes at waypoint #2 and one enemy ship each at waypoints #3 and #4. He doesn't know what is going to happen but he can see enough to be prepared for some sort of action at those locations.

The solution to not giving away your mission's secrets is to hide the enemy's location from the player.

On the left side, near the bottom of the main Mission Builder screen, Is a check box marked "**Hidden to player**". If you click this box the selected object will not be visible on the Advanced Info screen. The object will show up in the mission just as it is supposed to but it will no longer appear on the Advanced Info map. You can make any object, aircraft, ship, or infrastructure that you can place in a mission hidden to the player except the player's aircraft. It makes sense then, that in most cases you should be hiding the enemy from the player during the briefing. Sometimes the mission is to destroy a specific target or you have reconnaissance photos that show the location of certain objects. In those instances you may want to leave those objects visible in the briefing. But as a general rule, you should hide the enemy from the player.

To hide an object that is currently visible, click on the icon for that object. For a moving object such as a plane or ship, you can click on any waypoint. With the object highlighted click the "**Hidden to player**" box. When you do this you will note that an "**X**" will appear on top of the object's icon and all of its waypoints. This is how you can tell, looking at the mission map, which objects have been hidden and which have not. If you'll page back up to the title page for Part Two you can see a mission map where all of the enemy objects have been marked as hidden. Notice the "X's" on each object or waypoint. The friendly planes are not hidden.

Let's hide the Zeros and patrol boats in *Luck of the Draw*. Click on the icon for either of the two Zeros then click the "**Hidden to player**" box. Do the same thing for the other Zero then for each of the two patrol boats. As you hide them you should notice that a red "X" appears on each icon and all of their waypoints. Now when they go into Advanced Info this is what the player will see:



This is much better. We have removed all sign of the enemy from our map. As good as this is however, it isn't perfect. A really experienced player will note that waypoints #4 and #5 are set very close together. Also, while the briefing states that we are to fly a triangular pattern over Iron Bottom Sound, there is a kink in our triangle at waypoint #3. For the experienced player these are hints that something could be happening at these locations. Let's further mask our intentions.

Mission Builder allows you to hide an individual waypoint as well as hiding the entire object. Click on the player's waypoint #3 and go to the Route Tab. Near the bottom, just below the "Warp to" box is a box marked "**Hidden**". Click this box to hide waypoint #3 on the Advanced Info map. Then go to waypoint #5 and hide that waypoint as well. Remember, to hide the entire object and all of its waypoints click the "**Hidden to player**" box on the Flight Tab, to hide an individual waypoint, click the "**Hidden"** box on the Route Tab.

Notice that nothing changed on the Mission Builder map when you hid the waypoints. There was no little "X" or any other indicator on the map itself. The only way you can tell an individual waypoint is hidden is to look on the Route Tab.

Now let's take a look at what our player will see when they go to Advanced Info:

The final Advanced Info map as seen by the player

This is perfect. Now the Advanced Info map is consistent with our triangular flight path and there is no indication whatsoever of trouble ahead.

Use the "hidden" features to polish off your missions. It's self defeating to design a challenging mission only to tip your hand as to its contents when looking at the Advanced Info briefing map.

We'll make a few final enhancements to our mission then package it up. Let's start by adding our debriefing back in.

Create a new trigger named "**Player arrives at WP#6**". Select your aircraft and use these parameters:

'Type' - 'Status'
'Param' - 'Waypoint Reached'
'Operator' - 'Equals'
'Value' - '6'.

Open a new event and name it "**Mission Debrief**". Select **Player arrives at WP#6** as your trigger. Your action will be "**Debrief text**". In the text window type in: **Those Zeros caught you off guard but you handled yourself well. Good work!** Save this event then create another new event.

Name this event "**Patrol Boat Debrief**". For your first trigger select the event "**Warp at WP#3 – Boat**". Then click the "**OR**" button and select the event "**Warp at WP#4 – Boat**" as your second trigger. Notice that when you do that the second trigger appears in a second "In use" box below the first and that the "OR" button is shaded. We are using these two events as our triggers because they contain both the spawning of the patrol boat and its being sunk. See the illustration on the next page.

Either/Or

Up until now we have used a single trigger or group of triggers to activate an event. When all of our triggers are "True" the event occurs. By clicking the "OR" button it allows you to use a second trigger or group of triggers to activate the same event. Now the event will occur if **EITHER** the first set of triggers is true **OR** if the second set of triggers is true. Creatively used, this can open up many more possibilities for triggering events.

Your action will be "**Debrief text**" and in the text window type: **Nice job bagging that patrol boat.** We don't know if a patrol boat will spawn in the mission or not, but if one does and you sink it you will get this debrief message in addition to the one we entered above. Close out this event.

Unfortunately we cannot test our debrief messages until we fly the mission in the game. As you've already seen, when you fly a mission from within Mission Builder it exits back to the mission map when the mission ends but before displaying the debriefing data. Before we do that let's clean up our briefing information.

Open the Mission Setup window and change the first sentence of the Overview to read: **You and a wingman are flying a triangular grid out of Henderson. Keep alert and destroy any enemy targets that you encounter. The mission will end when you return safely to Henderson**. The overview will now accurately reflect the fact that we have a wingman and is sufficiently vague not to tip our hand. Save your revised mission and exit Mission Builder. Now is the time to check the full mission. Start up CFS2 and go flying. You might want to fly it several times to allow some of the random options to kick in. Check the Advanced Info map to be sure you properly hid all of the enemy objects and our two waypoints. Also check the debrief screen for accuracy. If anything is not consistent with the Handbook, go back over that section and correct your mistakes.

Summary

As we saw in this chapter, by adding percentages to trigger events, even a very simple mission can be beefed up and made more interesting. Using random chance isn't for every event, and may have no application at all in an historical mission where you are trying as much as possible to replicate an actual event. However, in fictional missions it can add a lot to the experience. When multiple missions using random chance are combined into a campaign the number of variations in how the campaign will play out can be almost limitless.

Any event can be triggered by random chance. In our example we used percentages for spawning enemy targets but you could also use it for:

- spawning flights of friendly planes, ships, vehicles
- activating an engine fire
- playing a sound
- changing waypoints (alternate route)
- displaying text
- unhiding a goal

Be creative in applying percentages to your missions.

Hiding the enemy objects from the player is something that almost any mission can benefit from. As you saw in our examples, when left unhidden the player is given a road map of what to expect. Hiding the enemy gives you, the designer, the edge and helps keep the mission a mystery. If you want to create exciting, fun, and challenging missions use all of the tools Mission Builder provides in combination to your advantage.

Reconnaissance Photos

You can add reconnaissance photos to any mission you create. The game comes with 14 recon photos but these are specific to the campaigns and may not be of any benefit to you in your custom missions. However, you can create and add your own photos to the game very easily.

The stock recon photos are in jpeg format, 498 x 374 pixels, and 72 dpi. After much experimentation, I've found that you will get the best results if you stick to these same criteria. Any jpeg image can be used; photos, map images, drawings, even screen captures. Although you can use color, you will get better results if you convert the image to greyscale first. Use any photo editing program you like to convert the image to the specifications above. Give the photo a logical name and save it in the **PHOTOS** folder inside your main **Combat Flight Simulator 2** directory.

To insert your recon photos into the mission, go to the Mission Setup screen and click the **Add** button at the bottom of the recon photos window. You will see a list of all of the photos located in the PHOTOS folder. Highlight the one you want to use and click **OK**. You can add up to four photos per mission.

Chapter Six – avoiding the never ending mission

Preflight
If you download missions from the Internet, eventually you'll come across two of my pet peeves, the mission that ends too soon and the "Never Ending Mission". Both are examples of poorly designed missions, but both are easy to avoid

The mission that ends too soon is usually one where the designer failed to take control of warping. The basic scenario in this mission is that you take off, warp to a location and get another blue warping message. Not knowing differently you press "X" and find yourself back at your base, lined up on final approach with the "Press X" to end mission message showing. Nothing happened! Where was the enemy? By failing to disable warp at the target area you end up missing all of the action. Fortunately this error is easy to avoid. By taking charge of warping as we've shown you in this Handbook you learn when to turn it off and turn it back on. You, as the mission designer, control the action and keep the player on course.

The never ending mission is just the opposite. This is a mission where you get stuck out there somewhere over the Pacific and can't get home. The mission just won't end until you press ESC, bail out, or crash. None of those scenarios are good because they all result in you failing to complete the mission. One cause for this is not turning warp back on. That's a simple designer error and easily fixed. Another cause for this is more complex. You can have a well designed mission such as *Luck of the Draw* and still leave a player stuck out there in cyberspace.

In *Luck of the Draw* we know the player will encounter a flight of Zeros (either two or three planes) and may encounter a patrol boat as well. Since warping is tied to the destruction of these objects, if for some reason the player doesn't shoot them down or sink them he can't warp away. The most frequent cause of this behavior is running out of ammo. If you have no means of fighting the enemy and you can't warp until he is destroyed you'll find yourself stuck. In a real combat situation, if you ran out of ammo you would disengage and head for home. You can design your CFS2 missions the same way by using alternate routes.

There are other reasons you may need to head back to the base early. Maybe you're low on fuel or damaged too heavily to continue. Both are good reasons for breaking off the fight and returning home; and both are easy to incorporate into your mission scenarios. In this chapter we'll explore using alternate routes as a means to return a player to his base and prevent the Never Ending Mission. We'll also show you how to use alternate routes to add another variable to your missions.

Incorporating Alternate Routes
Alternate routes can be added to any moving object, a plane, a ship, or a vehicle. However you can add only one alternate route per object. Effectively then, each moving object can have two potential routes assigned to it. An object always starts out moving on its primary route. The alternate route is invoked through the use of a trigger and an event. Let's begin by opening up *Luck of the Draw*. We are going to assign an alternate route to the player's aircraft as well as one of the patrol boats. Zoom in on the first patrol boat, the one at our waypoint #3, until you are at about the 15 mile width. Click on the patrol boat icon and then click on its Route Tab.

Near the top of the Route tab you will see two buttons, one marked “Primary” and one marked “Alternate”. The Primary button is the default and will be highlighted. Click on the “**Alternate**” button to activate it. When you move your cursor back to the map it will change to a ship icon. Wherever you place the cursor and click will be the first alternate waypoint for the patrol boat. The remaining waypoints are plotted just like any other moving object. Use the illustration below as your guide for plotting the alternate waypoints for this boat.

Note that a dashed line is used to designate the alternate route. Since your original course was hidden the alternate course will be as well.

With the alternate course plotted, raise the speed at each of these new waypoints to 30 knots, the maximum allowed.

The original course was to the southwest, away from Florida Island. Our alternate course is back to the northeast and at a higher speed. This will simulate the patrol boat turning around and racing back to the protection of the island.

We are going to link activating this alternate route to the patrol boat being fired upon. The easiest way to do this is to use ‘object health’ as our trigger. Technically, you could fire all your rounds and never hit the patrol boat and it wouldn’t change course. Then again, if you are that bad of a shot I guess he doesn’t have anything to worry about anyway.

Create a new trigger named “**Patrol Boat Hit**”. Be sure to select the correct torpedo boat then use these parameters:

‘Type’ -	‘Status’
‘Param’ -	‘Object Health’
‘Operator’ -	‘Less Than’
‘Value’ -	‘100’

With an object health of less than 100%, this trigger will be true as soon as there is any damage to the patrol boat. Close the trigger and open a new event.

Title this event "**Alternate Route – Patrol Boat**". Remember our lesson from Chapter Two:

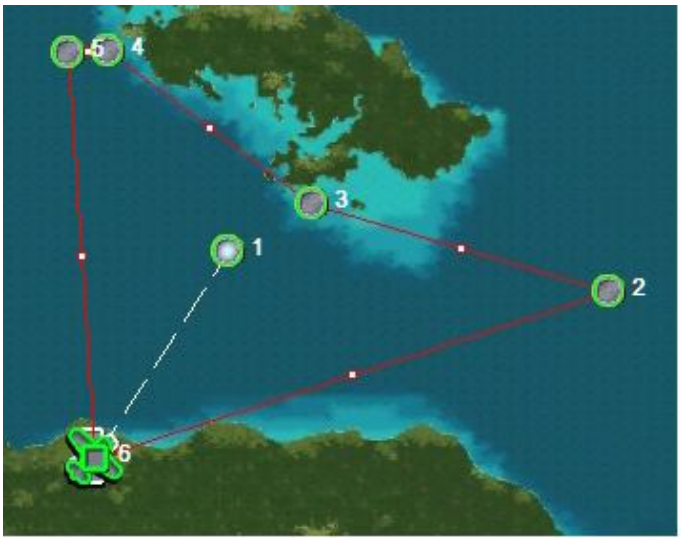
- An Object's Health equals '0' until it is spawned
- When an event is based on the destruction of a spawned object, always include spawning the object as one of the triggers.

Zero is certainly less than 100% so we must treat this alternate route event the same way we would an event based on the destruction of the patrol boat and include the spawning event as one of our triggers. If we didn't the game would apply the alternate route to the patrol boat from the start of the mission. For your 'In use' triggers then, select the event "**Spawn First Patrol Boat**" and the trigger "**Patrol Boat Hit**". Your action will be "**Change waypoint**". In the drop-down menu select the correct patrol boat and click the double right arrows "**>>**". <u>Be sure you select the correct boat!</u>

Now if this torpedo boat spawns and you are able to hit it, it will speed up, turn around and head back toward Florida Island. Don't expect miracles! Ships in CFS2 don't turn on a dime; they turn in a very broad arc. But she will turn and she will head for the island if you've plotted it out as we've shown here. (Of course you may well sink the patrol boat long before it completes its turn).

Next we'll set up an alternate route for our player and a couple of options for ending the mission. We want to give our player a way home in the event he runs out of ammo or is too badly damaged to continue so we will plot an alternate route back to Henderson. When setting your alternate route waypoints you must give consideration to what the scenario is you are trying to accomplish. In some missions your area of battle may be a long way from your airbase so you could place your first alternate waypoint almost anywhere along the line to your base and it would be OK. In *Luck of the Draw* we are flying a broad triangle and are never out of sight of Guadalcanal Island so we need to plot our first alternate waypoint carefully.

We don't know where our player will be when the alternate route kicks in (if it does at all) so we need to place our first waypoint in a logical spot that the player would fly through regardless of where he runs out of ammo. Look at the illustrations. On the left you can see that if we place our first alternate waypoint in the center of the triangle it is an unrealistic spot if our player runs out of ammo at either waypoint #2 or #4. He would not be flying the most direct route home. In this mission we need to place our first alternate waypoint closer to Henderson, as in the illustration below on the right. This is a much more logical point that the player would fly through regardless of where the alternate route is invoked.



Poor first Alt waypoint

Good first Alt waypoint

Using the illustration as a guide, highlight the player's aircraft, click on the Route Tab, and then click on the Alternate button and plot the alternate route. You'll notice that you cannot warp to the first alternate waypoint. Occasionally I want the player to actually fly to the first alternate waypoint, but most of the time I want them to warp to that spot. Since you can't warp to the first alternate waypoint, place the second alternate waypoint directly on top of the first one and mark it as "**Warp to**". Set your third waypoint on Henderson and select "**Land at base**" for it. Mark all three alternate waypoints as "**Hidden**". The default altitudes and speeds will be fine for our purposes. Now that we have an alternate route plotted for our player if needed, let's create some triggers and events to use it.

Open a new trigger and name it "**Out of Ammo**". Select the player's aircraft and use these parameters:

'Type' - 'Stores'
'Param' - 'Gun Rounds'
'Operator' - 'Equals'
'Value' - '0'

Close it and create another trigger named "**Player Damaged**". Select the player's aircraft and use these parameters:

'Type' - 'Status'
'Param' - 'Object Health'
'Operator' - 'Less Than'
'Value' - '90'

> ☛ You can set this damage value to anything that you'd like. Setting it at 100% would mean that if you got even one hit on your plane you would be ordered home. That isn't realistic. I've found that setting the damage value at between 80 and 90 percent is fairly realistic, but it will vary somewhat from plane to plane. Don't set it too low or you won't be able to make it home. Again, focus on the technique, not the numbers.

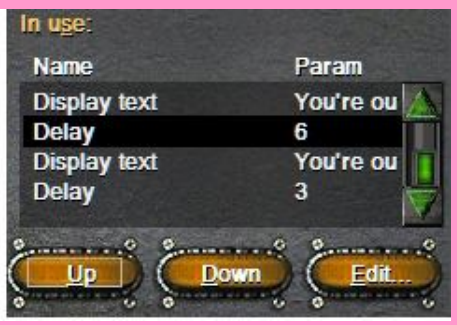
We will use these two triggers to activate our alternate route if needed. I sometimes create a third trigger for low fuel. However, in this case we are so close to Henderson that running out of fuel is not likely so we will stick with just these two triggers. After finishing the second trigger open a new event.

Title this event **"Alternate Route – Ammo**". For your trigger select "**Out of Ammo**". Your first action will be "**Change Waypoint**". Be sure to select the player's aircraft. Then select "**Display text**" and type in "**You're out of ammo! Break off and return to Henderson**." Insert a "**6**" second delay then repeat the text, followed by another **6** second delay and repeat the text again. Finally insert a "**3**" second delay then select "**Miscellaneous**" and "**Enable warp**".

> ☛ There is an easy way to repeat text such as we're doing in this event.
> - Enter your first Debrief text as described above. Then copy the text, enter a second debrief text, followed by a third text (or however many you are going to use).
> - Next enter your first time delay, then enter the second, then the third.
> - Use the **up** and **down** buttons to move the delay times in between the text messages.
>
> It is much faster to enter all of your repeated text followed by all of your time delays then simply move them up or down to stagger them properly. It sure beats going back and forth and retyping each time!

All 3 texts

All 3 delays



After moving UP or DOWN

Create another event named **Alternate Route – Damage**". For your trigger select "**Player Damaged**". Your first action will be "**Change Waypoint**". Be sure to select the player's aircraft. Then select "**Display text**" and type in "**You're too badly damaged to continue! Break off and return to Henderson**." Insert a "**6**" second delay then repeat the text, followed by another **6** second delay and repeat the text again. Finally insert a "**3**" second delay then select "**Miscellaneous**" and "**Enable warp**".

You now have two trigger and event combinations that will return the player to the base in the event he either runs out of ammo or is damaged too heavily to continue. No more wandering around following waypoints when you can't fight back. Congratulations, you've avoided the dreaded Never Ending Mission!

Open a new trigger and name it "**Player arrives at ALT WP #2**". Select the player's aircraft and use the following parameters:

'Type' - 'Status'
'Param' - 'Waypoint reached'
'Operator' - 'Equals'
'Value' - '**-2**'

Note the value is **MINUS TWO (-2**). The minus sign tells Mission Builder that you are referring to an alternate waypoint. The default numbers are designated with a plus sign (+) and you don't need to enter the plus sign when referring to a primary waypoint, but you must enter the minus sign to designate an alternate waypoint.

Open a new event and name it "**Warp at ALT WP#2**". Your trigger will be "**Player arrives at ALT WP #2**". **Disable warp**, set a **5** second delay, then **enable warp**.

At this point you should test fly the mission to make sure the new triggers and events work. To test the ammo trigger you should wait until you get to one of the waypoints and simply fire your guns until you are out of ammo. You can't really test the damage trigger very easily, but if you set it up identical to the ammo event, except for using the damage trigger, it should also work.

Remember one important thing. CFS2 will not allow you to warp if there are enemy planes in your area. If you run out of ammo before you destroy both Zeros you will see the message but you can't warp until they are either destroyed or you fly far enough away from them. In this mission it isn't too big a deal since you are within sight of Guadalcanal and can turn for home and out run them. In missions where my player may be hundreds of miles away from the base I will include a heading message along with the break off message. My text might read something like "**You're out of ammo! Break off and return to Midway on a heading of 117°**". This way the player knows which way to turn to get away from the enemy.

Finally, in a "There and Back" mission where you fly to some distant point, engage the enemy, and return home via the same basic route, you don't even need the alternate route to avoid the Never Ending Mission. Just build in your triggers and events for ammo, fuel, and damage, and give the break off message and heading as described above. I like to delay about 90 seconds then enable warp under this circumstance. This gives the player time enough to fly out of range of the enemy fighters and out run any pursuers. Once out of range, warp will be enabled and the player heads home via the primary route.

Landing Back at the Base as a Mission Goal

We'll finish off this mission by requiring the player to land safely back at Henderson and make that a mission goal. We get into the habit of pressing "X" to end our missions when we return to our airbase, but occasionally it is fun and challenging to require the player to actually land his plane to end the mission. We can build triggers and events to incorporate this action as a mission goal.

First we must define what landing safely back at the base means. We'll choose:

1. the player arrives at Henderson (defined as reaching the last waypoint)
2. the aircraft is stopped
3. the engine is off

By now you should be proficient at writing triggers so create three new triggers, choose the player's aircraft each time and set these titles and parameters:

Player arrives at ALT WP#3

'Type' - 'Status'
'Param' - 'Waypoint reached'
'Operator' - 'Equals'
'Value' - '**-3**'

Player Stopped

'Type' - 'Status'
'Param' - 'Speed'
'Operator' - 'Less than or equals'
'Value' - '0'

Engine Off

'Type' - 'Status'
'Param' - 'Engine On'
'Operator' - 'Equals'
'Value' - 'False'

Be sure the trigger for alternate waypoint #3 shows a minus three (-3). We don't need a trigger for arriving at waypoint #6 because we already created one earlier.

With your triggers in place open a new event and name it "**Land Safely at Henderson**". For your first set of "In Use" triggers select **Player arrives at WP #6**, **Player Stopped**, and **Engine Off**. Click "**OR**" then select your second set of triggers: **Player arrives at ALT WP #3**, **Player Stopped**, and **Engine Off**. Set a **3** second delay then display the text: "**You have landed safely at Henderson. The mission will now end.**" Then set a **6** second delay and select the miscellaneous action "**End mission**". Finally, click the box "**Count as mission goal**" at the bottom of the screen.

Create another event named "**Landing Instructions**". Select **Player arrives at WP #6**, click **OR**, and select **Player arrives at ALT WP #3**. Your first action will be "**Disable warp"** under miscellaneous. Then display the text "**You must LAND, STOP, and TURN OFF your engine to complete the mission.**" Set a **10** second delay then display the text again, followed by another **10** second delay, followed by the text again one last time. This is similar to the pattern we used when activating the alternate route.

There is a catch with this event however. In this mission we defined landing at Henderson as arriving at the last waypoint, engine off, and stopped. When these three triggers are "true" the event ending the mission occurs and we reach our goal. The problem is that once we reach that final waypoint we could crash and be killed and those triggers would still be true. Plowing into the ground will stop you and turn off your engine, trust me. You could be killed, yet still achieve the mission goals. There are ways to compensate for that by designing even more complex triggers involving surviving for a certain period of time after meeting the other triggers, etc. I'll let you experiment on your own if you want to work on that. Personally I'm satisfied that most of the time the player will set her down just fine!

We have now made it a goal to land safely back at our airbase. Regardless of whether the player arrives back at Henderson via the primary or alternate routes he will be instructed that he must land to complete the mission. Once he is safely on the ground and the engine is shut down the mission will end automatically. By disabling warp at the final waypoint, we have eliminated the "Press X" to end mission message that normally appears. Once again, by controlling warp you can control the flow of the mission; in this case by adding the goal of landing at our base.

☛ A variation of this would be to let the player land anywhere but still use the safe landing to end the mission. After the player completes whatever it is you want him to do, disable warp then display a message similar to this: "Mission accomplished. You may land at any friendly base. The mission will end when you are safely on the ground". Use **Player Stopped** and **Engine Off** coupled with the event that displays this message as your triggers. Now, regardless of where he lands the mission will end.

Save this mission as "**Alternate Reality**". Be sure to give it a different file name, then either exit Mission Builder and fly it from CFS2 or click the "**Fly Now**" button and test your new mission ending.

☛ You can use this same technique for making carrier landings a mission goal. You can't use stopped as a trigger because the carrier is moving. When I set up a carrier landing I use arriving at the last waypoint, engine off, altitude and speed.

I get the altitude while sitting on the deck of the carrier, then create a window using two triggers (AGL greater than and AGL less than). That compensates for the difference in some carriers. I also use a trigger with a speed less than normal flying speed but greater than the carrier speed. For example, when the player is at an altitude of between 50' and 70', traveling at less than 50mph, has reached his final waypoint, and the engine is off, I trigger the event to end the mission. You can also build this event without the altitude, using speed alone.

I also use landing a seaplane as a mission goal. Again, I use reaching the last waypoint, engine off, and stopped. To add to the challenge, create an Event Area close to a wharf or dock and require the player to taxi until they are within the event area, then instruct them to shut down their engines and end the mission.

Other Uses for Alternate Routes

Alternate routes can add another dimension to your missions. In our example we used alternate routes in a reactive way (after we hit the torpedo boat, after we ran out of ammo, etc.). You can also use them in a proactive way, to change the mission assignment for example.

Try setting up a mission where the player is given both a primary and a secondary target during the briefing. Inform him that he will be notified which target to strike as he nears the target area. Plot one target using the primary route and the other using the alternate route. As the player approaches the

target you can use a random percentage trigger to determine which way he goes. You can dress up a mission like this by adding a flight of AI planes and assigning them an alternate route as well. Use one trigger to invoke both alternate routes and you can switch to the secondary target while the AI flight turns toward the primary.

You can further improve the mission by playing the sound **t039, "***Home plate calling flight leader. Stand by for new mission orders*." first, and then display text ordering the player to the new target. Another technique is to give your player a routine CAP mission and as he reaches the final waypoint set up the landing trigger as shown above. Set an event that is triggered by the player reaching the last waypoint and being below 500'. When the player gets to this point invoke the alternate route and send him off to intercept the enemy at some other location. When I use this method I add additional triggers that check to see if the player left his gear and flaps down. If so, I use radio messages to order him to raise them.

Alternate routes are another tool in building interesting and challenging missions. You won't necessarily use them every time but they can be very helpful in preventing the Never Ending Mission or in just adding variety to your flying.

☛ Here are a couple of things to keep in mind regarding waypoints:

1) You're limited to 32 waypoints for each moving object (airplane, ship, or vehicle). That means you'll need to finish whatever it is you're doing within those 32 waypoints. Thirty-two is a lot, but sooner or later you'll come up with some hare-brained idea and run into the limit (I know from experience). One way to beat the system is to use an alternate route. You get an additional 32 alternate route waypoints per object. So if you really need to stretch things out, you could plot your first 32 waypoints then up to 32 more using an alternate route. Create a trigger/event that invokes the alternate route when you hit the 32nd primary waypoint and you're in business.

2) When you warp, the game computes the time and distance to the waypoint you are warping to as though you had flown it in real time via all of the intermediate waypoints. Look at this illustration. This is a diamond shaped flight plan out of Midway. The first and last waypoints are only 20 miles apart. If waypoints 2, 3, and 4 are not marked as "Warp to", when you press "X" you will jump to your landing approach at Midway. Although only a second will have passed in real time, 3 hours and 41 minutes will have run off the game clock and you will have flown 666 miles! You may have a devil of a time making the trip if you were low on fuel.

Chapter Seven – customized debriefings

Preflight

By the time we finished *Alternate Reality* we've come up with a variety of ways for the mission to play out. Our player might complete the full mission and return safely, he might run out of ammo and need to divert for home, or he may get so badly shot up that he has to break off and return to Henderson. In each case he may or may not shoot down all of the Zeros and sink a patrol boat. You could write one blanket debriefing that generically covers the mission, or you could be creative and customize your debriefings to how each mission ends. In this chapter I'll show you how to craft debriefings that are more specific to each situation. Personalized debriefings are more fun and add an air of realism to your completed missions.

Matching your Debriefings to the Mission Ending

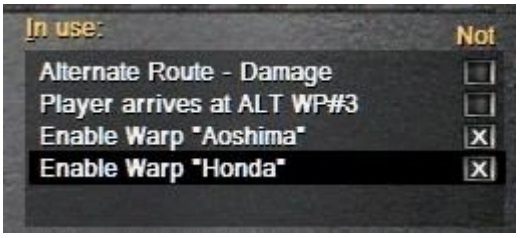
Begin by creating one final trigger. Name this one "**Player Bails Out**". Select the player's aircraft and use these parameters:

'Type' -	'Status'
'Param' -	'Pilot bailed out'
'Operator' -	'Equals'
'Value' -	'True'

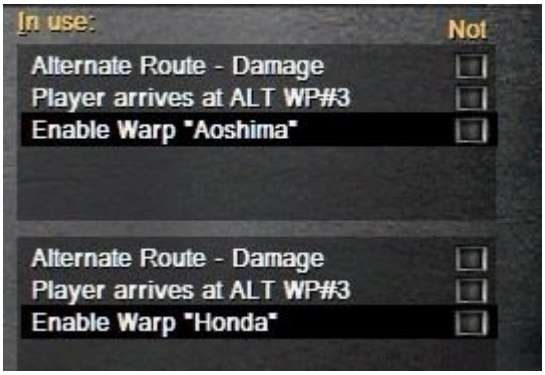
We already created two debriefings, one connected to bagging the Zeros and one to sinking a patrol boat. Both of those debriefings are fine and won't need any modification. We will focus on debriefings connected with problems that might crop up during our mission.

Open a new event and name it "**Player Bails Out**". Select the trigger by the same name that you just created. Select **Debrief text** as your action and type in this text: **You lost your plane! We're glad you survived, but those Wildcat's don't come cheap. You need to be more cautious when engaging enemy targets. Get dried off and head over to the Officer's Club.** We're making two assumptions here, one that the player survives (in a small percentage of bail outs your chute won't open) and two, that he lands in the water. We'll be right over 90% of the time making our debrief seem more personal. Next we'll create debriefings for running out of ammo and being damaged. In each case these events could occur either before or after we have shot down the Zeros. We'll make debriefings both ways.

Create a new event titled "**Debrief – Damage – No Zero**" Select the following triggers "**Alternate Route – Damage**", "**Player arrives at ALT WP #3**", "**Enable Warp "Aoshima**", "**Enable Warp "Honda**". Check the "**Not**" box for both "Enable Warp Aoshima" and "Enable Warp Honda". For this event to occur the player must be heavily damaged, reach the base (ALT WP #3) and neither flight of Zeros has been destroyed. That means, this event can only occur if the player gets shot up by one of the flights of Zeros and escapes. Select **Debrief text** as your action and type in this text: **You're lucky to be alive! You were badly damaged battling those Zeros and almost lost your plane. You need to watch your six more and fly more defensively. We can't afford to lose any more good pilots.**



☛ By now you know you can simply copy these debrief texts from the Handbook and paste them into your mission's debrief text box.



Next, create an event titled "**Debrief – Damage – Zeros**" For your first set of triggers select "**Alternate Route – Damage**", "**Player arrives at ALT WP #3**", and "**Enable Warp "Aoshima"**". Then click the "**OR**" button and select "**Alternate Route – Damage**", "**Player arrives at ALT WP #3**", and "**Enable Warp "Honda**". Select **Debrief text** as your action and type in this text: **Nice job bagging those Zeros but you got so badly shot up today we'll have to scrap your plane. Try to be more careful when engaging the enemy. Don't take foolish chances. Next time you might not be so lucky.** This event will activate if the player is sent back to Henderson after either of the flights of Zeros are shot down. We don't know when he will reach the damage threshold. It could be during the fight with the Zeros or it could be after engaging a torpedo boat. Either way our debrief text is still applicable.

In each of these events we use the "Enable Warp…" event as one of our triggers as that event covers both the spawning and the destruction of the Zeros. Next we'll create two more events similar to these based on running out of ammo.

Create an event titled "**Debrief – Ammo – No Zero**" Select the following triggers "**Alternate Route – Ammo**", "**Player arrives at ALT WP #3**", "**Enable Warp "Aoshima**", "**Enable Warp "Honda**". Check the "**Not**" box for both "Enable Warp Aoshima" and "Enable Warp Honda". This event will only occur if the player has run out of ammo while fighting the Zeros. Select **Debrief text** as your action and type in this text: **You ran out of ammo during a dogfight! You're lucky to be alive. You've got to be more careful up there. Get in close, be sure of your target and use short bursts.**

Finally, create one last event titled "**Debrief – Ammo – Zeros**" For your first set of triggers select "**Alternate Route – Ammo**", "**Player arrives at ALT WP #3**", and "**Enable Warp "Aoshima"**. Then click the "**OR**" button and select "**Alternate Route – Ammo**", "**Player arrives at ALT WP #3**", and "**Enable Warp "Honda**". Select **Debrief text** as your action and type in this text: **Nice job bagging those Zeros but you used up so much ammo you ran out later when it was needed. You must learn to conserve your gun rounds. Get in close, be sure of your target and use short bursts. Now head over to the Ops tent for debriefing.** This debrief will be seen only if the Zeros are destroyed and the player runs out of ammo later, in all likelihood fighting a torpedo boat.

Our player will now see one of seven different debriefings depending on how the mission plays out. Granted there are some odd ball combinations of events that we didn't cover, but those are rare and you've learned the basics of customizing a debriefing to a specific mission ending. While you may not use this many debriefings in every mission, customizing a debrief gives the mission a realistic look and a more personal flair. Save this mission as "**A Brief Hop Around Iron Bottom Sound**". Exit Mission Builder and fly it in CFS2 to test it. Try to create the situations we built events for and see how many of the debriefings you can trigger.

☛ The ammo triggers will work even if you have your settings at "Unlimited Ammo". The trigger is based on the round count. When the count reaches zero the trigger activates, even though the guns will continue to fire. Try it. Select unlimited ammo, go into HUD mode and watch your ammo countdown to zero, you'll still get the 'out of ammo' message.

Limitations on Briefing and Debriefing Texts
There is a size limitation for both the briefing and debriefing texts in Mission Builder. I don't know how big exactly, but if your text stretches to a couple of paragraphs you are getting close. If you exceed the limit, Mission Builder will crash to desktop (CTD) and you will lose everything you've added to your mission since your last save. That means, if you haven't saved versions as you've gone along, you will lose the entire mission and that could represent a lot of work.

There is a workaround. You can hand edit the mission and add the longer text that way. Simply type up your text in Word, Notepad, or whatever text editor you like to use, and copy and paste the information into the appropriate spot in the mission file. Not only does this allow you to have bigger briefings but it also allows you to spell check your work first. Several authors who write historic missions use this technique to add detailed, historically accurate, information to their briefings and debriefings. We'll cover hand editing missions in Part Three – Enhanced Mission Building II.

One caveat to adding the longer text this way, if you open the mission in Mission Builder and try to edit any portion of the longer text, Mission Builder will CTD.

Debrief messages are events triggered like any other event, they will occur when all of the "In use" triggers you selected are true. That means you can have multiple debrief messages appearing in the same mission. They will be packed one after the other and can be rather confusing. We took advantage of that with our short, generic debrief dealing with the patrol boat, but sometimes multiple debriefs can be in conflict with one another. If that is happening to you, go back and review your triggers for the debrief events in question and follow the logic to figure out why both are appearing. To solve the problem you may need to use different triggers, or you may have to drop one of the debriefs completely.

Summary
Over the last three chapters I've showed you how to take a simple mission and enhance it into one that is less predictable and more personalized. Not every one of the techniques we've covered is right for every mission. However, knowing how to incorporate them and the other techniques in this Handbook will give you more options when designing your own missions. The more you know about Mission Builder, its abilities and limitations, the better your finished product will be. Whether you're designing missions solely for your own use or planning to upload them to the Internet, you'll want them to be fun and challenging.

We expanded *Beginner's Luck* from just four triggers and four events to nineteen triggers and twenty-three events, yet we have barely scratched the surface of what's available in Mission Builder. In the next chapter we'll look at designing a mission without any waypoints, controlling all of the action through the use of event areas and on-screen instructions to the player.

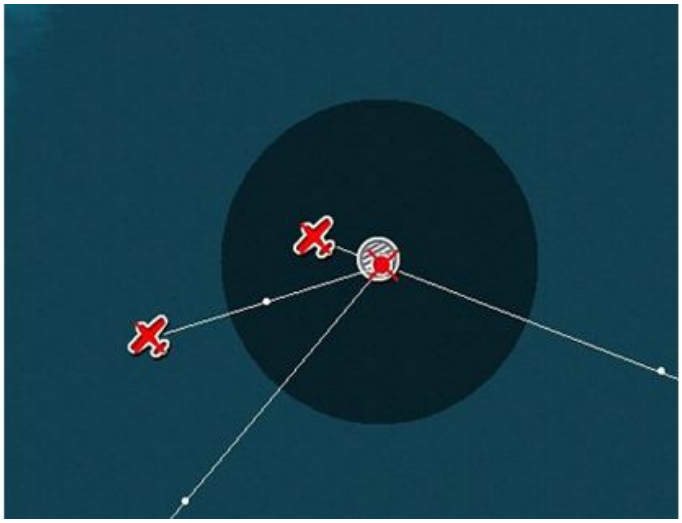
Chapter eight – missions without waypoints

Preflight
Some players don't like to use waypoints. They prefer to fly by a compass heading and not be restricted by the structure of hitting waypoints. You may have noticed that when a waypoint is tied to an event and you don't warp to the waypoint, you must fly within about a quarter mile of the waypoint marker to activate the event. For some players that's just too restricting. In this chapter I'll show you how to design missions without using waypoints or by using waypoints only to facilitate warping.

Avoiding Using Waypoints Altogether (Real Time Missions)
You can essentially eliminate all of the player waypoints from your mission. No waypoints means no warping, which means you'll fly the mission in real time. You can still spawn ships, planes and vehicles; either by using Event Areas to trigger the spawns or by delaying the appearance of the object in the mission for a set number of minutes. You could also eliminate spawning altogether and simply plot the courses in such a way that the player and the object arrive at the same place at about the same time. Finally, you could combine all of these techniques in the same mission if you wanted to. I'll demonstrate how to use event areas to control the spawning and flow of the mission.

For this example we'll look at how our mission *Luck of the Draw* could be written without waypoints. You can't actually eliminate all of the player waypoints because the game requires a minimum of two waypoints for any moving object. However, you can effectively eliminate them by making the first waypoint "**Take off**" and placing the second waypoint directly on top of the first one and set it to "**Land at base**". You must set waypoint #2 to "Land at base" or you won't be able to turn off the "Press X to end mission" message. By setting it to "Land at base" you can disable warp and eliminate this message. You can also hide the second waypoint if you want to.

We would plot out all of the enemy ships and planes the same way we did in the original mission. Instead of waypoints at each potential combat location we would use event areas. With your Japanese planes and patrol boats in place, draw three event areas where our old waypoints #2, #3, and #4 were located. We'll names these event areas **"Area One**", "**Area Two**", and "**Area Three**" respectively. I chose to make the areas five, three and three miles in radius respectively. I also moved the Aoshima flight of Zeros until they were about a mile outside of Area One and in line with Henderson Field; this will still give them the effect of being on our tail should they spawn. See the illustrations.



We have the same mission only we have eliminated the player's waypoints. The player will now fly the route in real time and we must give him clear instructions what direction to fly in so that he triggers all of the action. With that in mind you need to be careful when choosing how large to make your event areas. If you make them too small the player will miss them and fly off into the sunset. If you make them too big the action may not be close to where the player enters the event area. Trial and error, and testing your mission will help you determine the best size.

We would use slightly different triggers and events to control the action. We would create one new trigger named "**Player at Waypoint #1**". Instead of triggers for arriving at waypoints numbers two, three and four we would create triggers named "**Player Enters Area One**", "**Player Enters Area Two**", "**Player Enters Area Three**". Your parameters in each case would be:

'Type' - 'Status'
'Param' - 'Area'
'Operator' - 'Enters'
'Value' - 'Area One' (substitute 'Area Two' and 'Area Three' as appropriate)

You should be able to see already that all we are doing is replacing our arriving at waypoint triggers with entering event area triggers. The rest of the mission plays out the same.

We'll need a new event named "**Disable Warp**". Your trigger will be "**Player at Waypoint #1**" and your action selections will be "**Miscellaneous**" and "**Disable warp**". You must always use an event like this to disable warp at the first waypoint on any mission flown in real time. If you don't, the game will keep flashing the "Warp to" message and, if the player presses "X", he would find himself lined up for landing at Henderson, causing the mission to end prematurely. Take control of warp to take control of your missions.

☛ If you are feeling comfortable with the Mission Builder, try modifying the mission *Luck of the Draw* instead of writing a new one from scratch. It might be a bit of a challenge but it will save you time.

Some of our events will be a little different than in the original mission. "Spawn Honda" and "Spawn Aoshima" are now controlled by the player entering Area One instead of warping there. Spawning of the patrol boats is now controlled by the player entering areas two and three. Construct those events exactly as you did in the original mission substituting the enter area triggers for the arriving at waypoint triggers. The trick is to get the player to those event areas so that the action can happen. We will use new events to accomplish that.

We'll start by creating another trigger named "**Player above 500 feet**". Select your plane and use these parameters:

'Type' - 'Flight'
'Param' - 'Altitude (AGL)'
'Operator' - 'Greater than'
'Value' - '500'

Now we need two events, one to tell the player what heading to take just as the mission begins and one that repeats that heading message after he gets airborne. Since we already have an event that disables warp at WP #1 we can simply tag our heading message onto that event. Highlight the event "**Player at Waypoint #1**" and click "**Edit**". Add an action delay of **20** seconds, then display the text "**Take off, climb to 3000' and turn to 71°**". Enter a **10** second delay followed by the text again

followed by another ten second delay and the text one last time. Now, while the player is starting his engine he will be instructed what to do and where to fly to.

Open a new event and name it "**Heading Instructions**". Use "**Player above 500 feet**" as your trigger and then enter a display text of "**Climb to 3000', turn to 71° and stay on that heading**". Enter this text twice with a **10** second delay in between. Now if the player misses one of our messages while he's busy taking off, he'll get a reminder when he climbs above 500 feet.

With these two events we have set our player in route to Event Area One and the first action. Now we need to move him along after battling the Zeros. In the original version we enabled warp after the Zeros were destroyed; now we will display a message giving him a new heading to take. Our old triggers were named "Enable Warp 'Honda'" and "Enable Warp 'Aoshima'". Our new triggers will be named "**New Heading 'Honda'"** and "**New Heading 'Aoshima'**". Our triggers will be the same but instead of enabling warp we will display the following text "**Turn to 286° and resume patrol**" followed by a **5** second delay then this text "**Hug the coastline of Florida Island but don't fly over land**". Repeat this sequence twice with a **10** second delay between the two sequences.

Your finished event actions should look like this:

Delay	5
Display text	Turn to 286° and resume patrol
Delay	5
Display text	Hug the coastline of Florida Island but don't fly over land
Delay	10
Display text	Turn to 286° and resume patrol
Delay	5
Display text	Hug the coastline of Florida Island but don't fly over land

If all goes well our player will return to his course, fly 15 miles or so and enter Event Area Two. Once inside Event Area Two a patrol boat may or may not spawn. We need to create two events to handle sending the player on his way; one if the patrol boat spawns and one if it doesn't. Much like what we did with the Zeros, we will build events like our original warping events but will give the player a new heading instead of enabling warp.

Our original events were named "Warp at WP #3 – Boat" and "Warp at WP #3 – No Boat**".** Our new names will be "**New Heading – Boat**" and "**New heading – No Boat**". The triggers will be the same as in the original but we will remove "Enable warp" as the action and replace it with displayed text. For the event with the boat use "**Turn to 308° and resume your patrol**", delay **5** seconds, repeat the text, delay **5** seconds again and repeat the text a third time. For the "No Boat" event we still have the **15** second delay followed by this text: "**Turn to 308° and continue your patrol but don't fly over land**". As with the other event, delay **5** seconds, repeat the text, delay **5** seconds again and repeat the text a third time. We now have covered the action at the old waypoint #3 and have sent our player on his way regardless of whether or not a patrol boat spawns at this location.

Finally, we need to direct our player home once he has taken care of business at Event Area Three. Our old events named "Warp at WP #4 – Boat" and "Warp at WP #4 – No Boat**"** will be replaced with events named "**Final Heading – Boat**" and "**Final heading – No Boat**". Remember, we're replacing the arriving at waypoint triggers with the entering Event Area Three trigger. The actions for both events will be the same: delay **5** seconds then display the text "**Turn due south and return to**

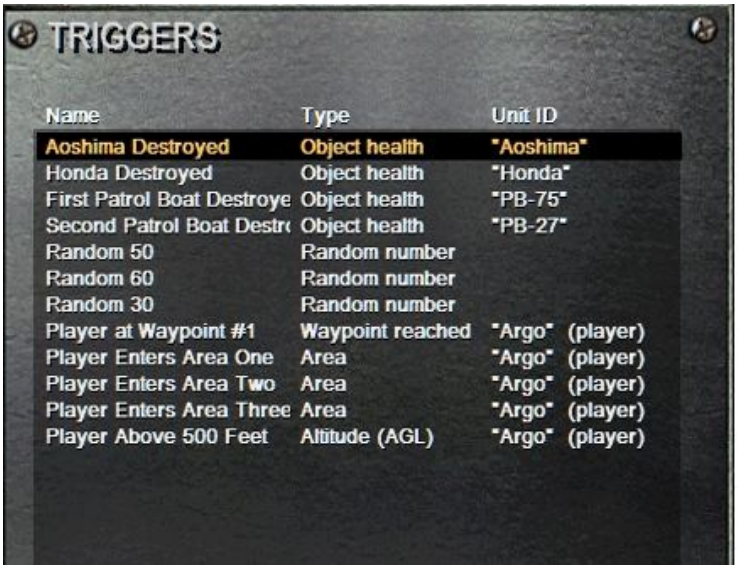
Henderson Field". Display this text three times with a **5** second delay between each as described in the previous paragraph.

The easiest way to end this mission would be to simply enable warp after leaving the second patrol boat (the old WP #4). That would return the player back to Henderson and a typical CFS2 mission ending. However, since we said no waypoints we'll end the mission another way. It will take the player at least five minutes to fly back to Henderson at full throttle so we will set up an event that ends the mission after about six minutes. That way he will either be close to Henderson or he may have already arrived and landed there.

Title this final event "**Mission Completed**". Your triggers will be "**Final Heading – Boat**", followed by clicking the "**OR"** button then "**Final Heading – No Boat**". For your action enter a delay of **360** seconds (six minutes), followed by the display text "**You have successfully completed your sortie. The mission will now end**." Followed by an **8** second delay then the **miscellaneous** action **End mission**.

We now have a full set of events that will guide the player along his route based on entering the various event areas and not by following waypoints. We have the same random chance of spawning the different Zeros or the torpedo boats that we had in our original mission and we have created a way to end our mission as the player flies back to Henderson.

Here are your final triggers and events. The order they appear in your mission may be different.



That completes the meat of the mission. We still need to tweak the briefing and debriefings but now would be a good time to test fly the mission and see if everything works the way you want it to. Save this new mission as "**No Way!**" give it a different file name and go flying.

It is not unusual with missions like this to have to tweak them to get them to run properly. If you are consistently missing one or more of the event areas you will need to either enlarge the size of the event area or change the heading instructions or both. There are some other problems associated with a "No Waypoint" mission. If the player goes into Advanced Info, this is what he will see:

Since there are no waypoints, the game zooms in to maximum close-up and all you see is a patch of dirt. If the player zooms out a little he still doesn't see anything that will help him understand the mission route:

Without waypoints and with all of the enemy objects hidden there is nothing for the player to see. You need to balance the value of the briefing map against having a mission with no waypoints.

With the mission tweaked to run the way you want it to, we will polish it off by modifying our briefing and debriefing texts. The "**Patrol Boat Debrief"** text we created in the original mission is still good to use in this mission. For our "**Mission Debrief**" we will use the same text but instead of triggering it with arriving at a waypoint, our trigger will be the event **Mission Completed**. With the events all finished, click on the **Mission setup** button and we'll add our briefings. Our original text is still good but we need to add special instructions to the player since we aren't using waypoints. Use these three briefings:

Overview: **You and a wingman are flying a triangular grid out of Henderson. Keep alert and destroy any enemy targets that you encounter. THIS MISSION IS FLOWN IN REAL TIME. TURN OFF YOUR TACTICAL DISPLAY AND FOLLOW ALL ON-SCREEN INSTRUCTIONS TO COMPLETE THE MISSION.**

Background: **Your plane was recently repaired and you want to take it out for a shakedown. You have planned a routine flight over Iron Bottom Sound.**

Intelligence: **Enemy activity has been at a low point lately. You should have an uneventful flight but watch for some Japanese patrol boat activity around Florida Island. THIS MISSION IS FLOWN IN REAL TIME. TURN OFF YOUR TACTICAL DISPLAY AND FOLLOW ALL ON-SCREEN INSTRUCTIONS TO COMPLETE THE MISSION.**

In missions of this type I like to give the player clear instructions regarding what they are to do. By turning off their tactical display they won't be confused by the waypoint marker always pointing back to Henderson. I use all caps to emphasize the instructions. There is a secondary effect of turning off the tactical display; the player must keep looking around to pick up the enemy targets. No radar on this mission!

That's one method of designing a mission without any waypoints. While it works, there are some inherent problems with the design. Let's look at designing a mission with only limited use of waypoints; one that tries to achieve the best of both worlds.

An Alternative Design

Here's another way to design a mission with no waypoints. Use lots of Event Areas of different sizes; some spawn enemy planes or ships while some spawn friendly planes or ships. Use random chance on some and not on others. In other words create lots of possibilities. Then just have the player take off and head for Florida Island. As soon as the first flight of enemy planes spawns and he gives chase, he will begin to enter different event areas, spawning other objects. Because you can never tell for sure where he will fly when chasing the enemy you create multiple, ever changing scenarios. In the illustration each circle represents a different Event Area.

Limited Waypoint Usage

As we learned, you can design a conventional mission without any waypoints but it does limit you somewhat. The two biggest problems that I see are no flight path on the briefing map and having to fly the mission in real time. Some players like to fly in real time, but I don't get any pleasure sitting in front of a computer screen for hours, looking at a simulated ocean and pretending that I am in a real cockpit. Personally I like a little warping to speed things along. You can easily construct a mission that uses waypoints to allow warping and display a course at the briefing yet still get the advantages of a no waypoint mission by limiting the use of your waypoints and having the player turn off their tactical display.

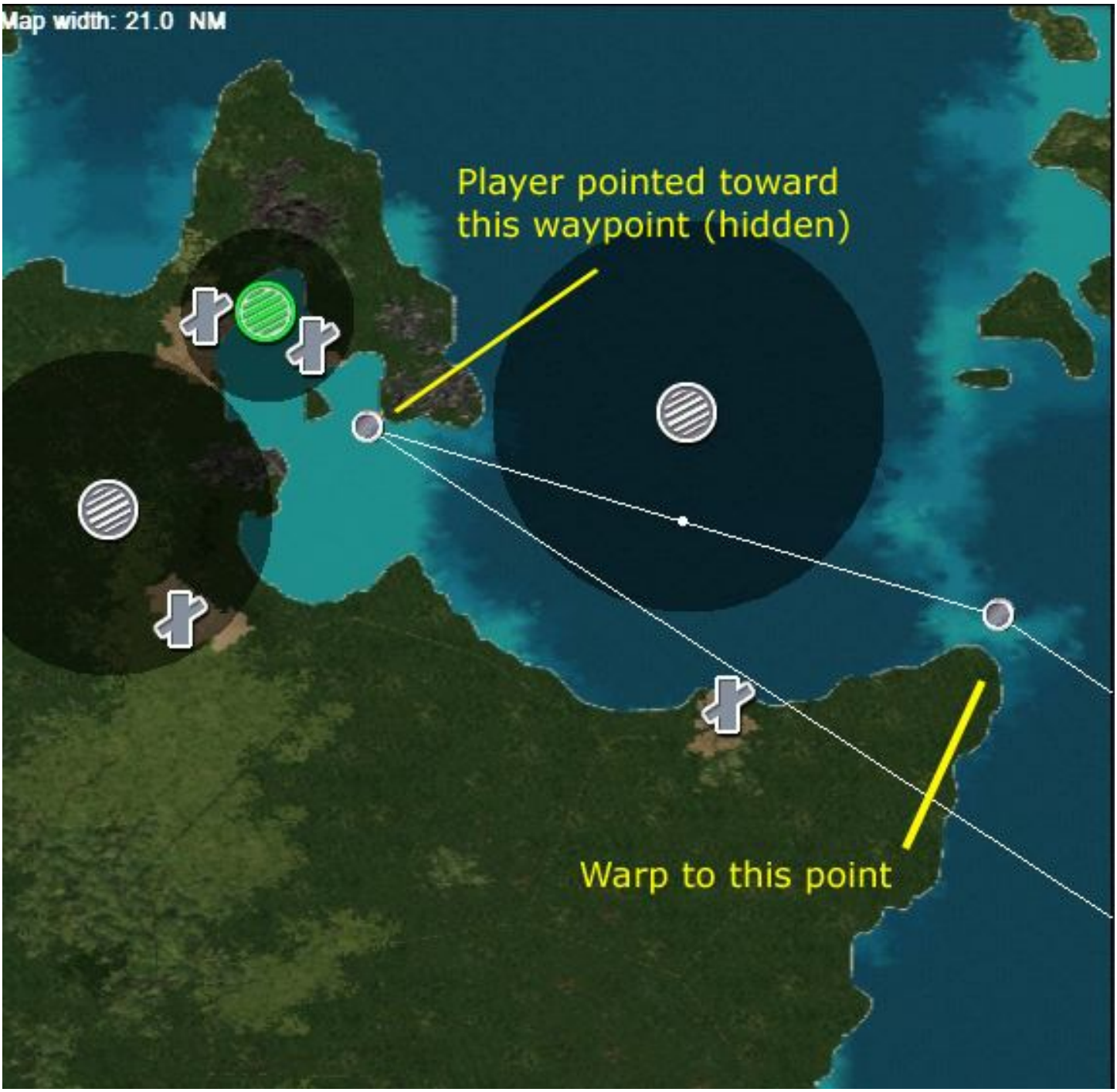
On page 91 you saw an illustration of what the player sees in Advanced Info – nothing! Compare that with the original briefing map from *Luck of the Draw* shown here:

It's easy to recreate this map in *No Way!;* Simply lay out four waypoints: "**Take off**", WP #2, WP #3, and "**Land at Base**". Place waypoints #2 and #3 inside of event areas One and three respectively. Don't mark any of the waypoints as "Hidden" as you want the player to see the course on the map. Also, set the action at waypoints #2 and #3 as "**Turn**". We still want our player to fly the mission in real time.

By setting the final waypoint to "Land at base" you have the option of having the player warp back to Henderson instead of flying back in real time. If he warps back to Henderson you could build in a landing module, requiring him to land, stop and turn off his engine to complete the mission. Our briefing instructions tell the player to turn off his tactical display; by doing that the player will see the course route on the briefing map but won't see any waypoint indicators while flying. You are now able to display a good flight map yet still fly the mission without waypoints, controlling all of the action via the event areas as originally designed.

Another situation where waypoints would be helpful is when the scene of action is a long way from the starting point of the mission. For example, say you wanted the player to take off from Henderson and attack the Japanese bases at Rabaul. That's a distance of over 550 nautical miles and it would take the player over three hours to fly there in real time. In that situation you might want to allow the player to warp to Rabaul, then disable warp and control the action through the use of event areas. When you're finished, you can allow the player to warp home. Again, if you instruct them to turn off their tactical display they won't see any waypoint indicators which could distract them from the actual course you intend for them to fly once reaching the target area.

Remember, when coming out of warp you are pointed toward the next waypoint. You would probably want to add an additional hidden waypoint at Rabaul so that when the player comes out of warp he is pointed toward the target and not back toward Henderson. See the illustration.



Summary

Writing missions without using waypoints, or with limited use of waypoints, is a novel way to approach your mission building. Like many of the techniques covered in this Handbook, it isn't appropriate for every mission. But the more you know about using Mission Builder, the more creative you can be in designing missions that are both fun and challenging. In the next series of chapters we will look at other elements of mission building and still more techniques to add to your bag of tricks.

Chapter nine – Moving Objects

Preflight

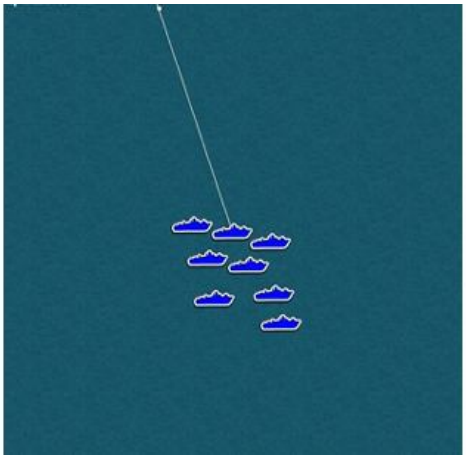
Moving objects consist of ships and ground units (vehicles). Like aircraft, you can assign primary and alternate waypoints for moving objects, create formations, spawn them, target them, and use them in triggers and events. In this chapter we'll explore the use of moving objects and cover some of their quirks and odd behavior. We'll also look at solutions and work-arounds for some of the most common bugs.

Ship Formations and Convoys

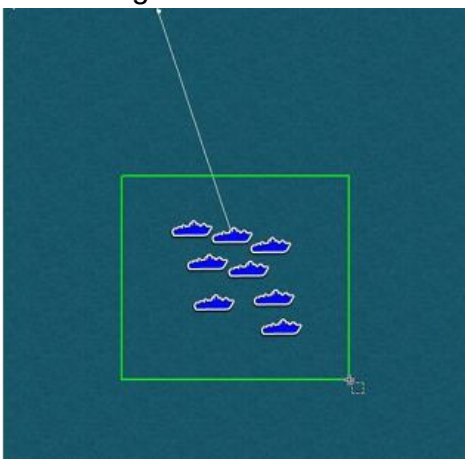
We used single ships in both *Beginner's* Luck and *A "Wake" Up* Call so you already have some experience with ships. You'll recall that you select a ship by clicking the "New Ship" button and choosing your ship from the drop-down menu much the same way as you do an aircraft. You have many of the same categories including 'Nationality', 'Unit ID', 'Skill', 'Goal', 'Start delay', 'Pre-damage', and 'Hidden to player'. In addition you have a check box for 'Invincible'. If you click on this box your ship can't be destroyed. The Route tab lets you set both primary and alternate waypoints, but your actions at any given waypoint are limited to "Turn" or "Do not evade". There are plenty of situations where you will want a single ship or a series of single ships to follow their individual waypoints, but there will be just as many times when you will want to build a convoy of ships all following the same course.

Formations (Mission Builder's name for a convoy) consist of a group of ships all following a single set of waypoints. Only one ship in the formation has multiple waypoints and all of the other ships follow this same course. To set up a formation begin by selecting a ship and laying out its course. This ship will be the Primary or Lead vessel. Then select the other ships you want to use and place them in the formation. Do not give these additional ships any waypoints, simply place the ship where you want it and right-click to position it. With all of your ships placed, hold down the **Shift** key and drag the cursor to form a box around the formation. Release the mouse key to highlight all of the ships then press **Ctrl + Alt + F** to create the formation. You will see a confirmation message.

Convoy route plotted

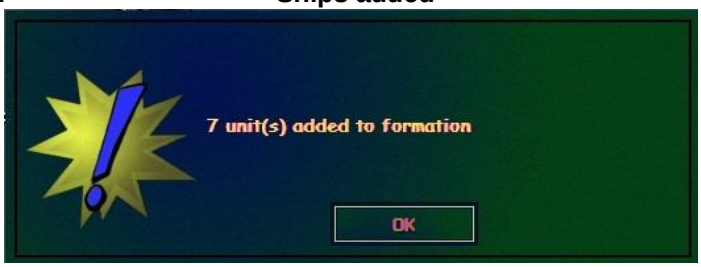


Ships added



Box drawn

Confirmation message



To add additional ships to your convoy simply place their icons near the others, box the entire convoy to highlight it and press **Ctrl + Alt + F** again. To remove an individual ship or a group of ships from the convoy, highlight the ship or ships and press **Ctrl + Alt + B**. Notice that when you do this the ships aren't deleted from the mission, merely removed from the convoy, so their icons will still appear and you will need to add additional waypoints for them if you plan to leave them in the mission.

There may be a limit to the number of ships you can have in a single formation but I haven't run across it yet. I've constructed convoys with over 100 ships and not had any problems. Realistically you probably won't have any convoys that big so size shouldn't be an issue.

One nice feature about working with formations is that you can move the entire convoy at once if you need to reposition it. Simply draw a box around the convoy to highlight it, (or select any one ship and press **Ctrl + Alt + X**) then drag any ship icon to move the entire fleet to the new waypoint #1.

Note in the illustrations that I moved the starting position for the convoy without moving the other waypoints.

☛ Give some thought to choosing your primary, or lead, vessel when designing your convoy. If the primary ship is sunk the rest of the convoy has lost its leader and will break formation and wander aimlessly. It doesn't look very realistic and creates a real mess, especially with large convoys. You wouldn't want to set the carrier in a task force as the primary vessel if one of the mission goals is to sink the carrier.

The primary vessel doesn't have to be in the lead of the convoy either. Any ship in the formation can be the primary. I recommend you select an escort ship or some other ship that is unlikely to be attacked and sunk. When setting your attack waypoints don't target the primary ship.

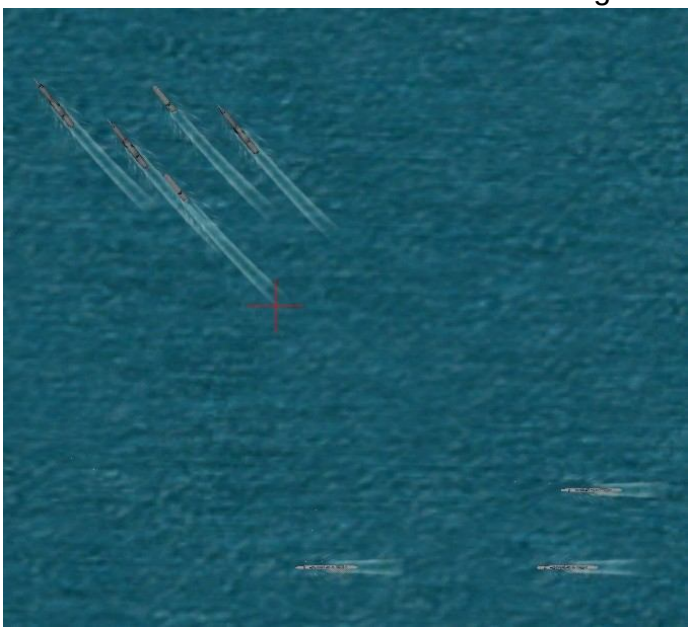
Sometimes, of course, your goal is to sink the entire fleet. In that case it won't matter and the confusion of the ships breaking formation may be just what you are looking for.

When plotting the route for a convoy, set all of the waypoint actions as "**Do not evade**". This will keep the ships in convoy formation even while they are under attack. If you select "Turn" instead, at the first shot the convoy will break apart like a covey of quail; ships going in all different directions, running into one another and generally looking like a big mess. It isn't very realistic.

There are a couple of well known quirks with the ships in CFS2, ships sailing over land and convoy ships not following their waypoints. Here are some fixes and tips for dealing with both of those issues.

Ships Sailing North

If you have a totally stock installation of CFS2 you may have noticed that the first time you constructed a convoy of ships, some of them turned and headed north instead of following their waypoints, something like you see in this illustration. This is the result of a bug in CFS2 that causes all **non-combat** ships to sail north. If you look closely the next time this happens you'll notice that it's the merchantmen, tankers, troopships, etc. that are turning while the combat ships hold to the course.



The problem is easily fixed by adding a gunstation to the ship's DP file. A DP file is the Damage Profile for the object and you'll find one in every ship and aircraft folder. It contains the information on the weapons carried by the ship and how much damage is needed to sink it among other things. Adding a gunstation to a non-combat ship is not difficult once you learn how to edit DP files, but there is a bit of a learning curve to mastering DP's.

Editing DP files is beyond the scope of this Handbook; however there is another way to fix the problem. Giovanni "Nanni" Cignoni is one of a cadre of designers who has made CFS2 great by offering free downloads of CFS2 add-ons. Go to his website, **Nanni CFS Page**, http://web.tiscalinet.it/Nanni/CFS/, and download his add-on **Nanni's Config – All Ships**. This file contains new configuration (CFG) files and DP files for all of the stock ships. In addition to solving the missing gunstation problem, he also improves the wakes, adds missing smokestacks, and does a number of other things to improve the look of the stock ships. Once installed, your non-combat ships will stay with the convoy and follow the waypoints just like they are supposed to.

☛ I strongly encourage you to get Nanni's ship add-on. An alternative to his fix is one provided by **Pen32Win.** He has a set of replacement DP files for download at www.sim-outhouse.com. These correct the "sailing north" problem as well as changing the way AI aircraft attack ships, pretty advanced stuff.

Whichever you choose, you'll notice that in your ship folder there is a **CDP** file. After installing the new DP file you **must delete the CDP** file for the changes to take effect. Don't worry; the game will create a new CDP file the first time the ship is used again. The CDP file is a special copy of the DP file that the game uses during play. If it isn't deleted after making changes to the DP file those changes will never be seen by CFS2.

Ships Sailing Over Land

Another quirk of CFS2 is that the ships seem to sail over land. Obviously, you won't notice this on the open ocean, but when you're close to shore you may see your destroyer turn and climb the beach. Not too realistic to say the least. Actually the ships are sailing **through** the land, not over it. Most of the small Pacific islands are set at zero elevation so the ships appear to be sailing on the surface of the land. When they hit a small hill you'll see that they remain at sea level and sail through the landmass. The game seemingly tries to compensate and pop them up as they move along keeping them near the surface. See the illustrations.

As you know, if you land your plane on the water it will sink, blow up, and you'll be killed unless you're using a ditching event as we did in *A "Wake" Up Call*. Unfortunately the designers of CFS2 did not apply a similar treatment to ships. When they hit land they just keep on going.

There is no fix for this problem but there are some things you can do to reduce the chances of it happening. First, avoid placing ship waypoints near land. That works for most operations but occasionally you want your ships or convoys to be close to land. In that case follow these three rules:

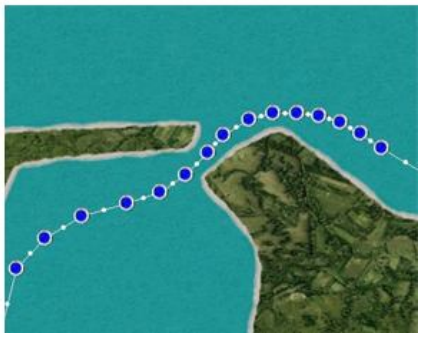
1. Always use '**Do not evade'** as your waypoint action. 'Do not evade' will keep your ships on course. If you use 'turn' when near land I can almost guarantee that they will run aground when they come under attack.

2. Avoid sharp turns – use more waypoints. As we noted in our mission *Alternate Reality*, ships in CFS2 do not turn quickly. Avoid sharp turns near land by using more waypoints. You've got 32 waypoints to work with, use them! A series of shorter, more gradual turns are easier for CFS2 ships to handle and will help you avoid having your ships run over an island instead of around it. See the illustrations on the next page for an example of setting up your waypoints and the correct results.

3. Slow down. Your ship is much more likely to take that corner at 10 knots than it is at 30 knots.

Poor route plotted

Good route plotted



Detail of passage

In the above illustration you see a poorly mapped route through a narrow gap and a good route. By using multiple waypoints you can avoid running over the land. This technique will work even when sailing through a gap like this, that is as narrow as our destroyer is long.

Approaching the gap

Threading the needle

Through the eye

Even if you follow these rules your ships may still run aground. But, by carefully plotting your waypoints and slowing down a little, you can eliminate the vast majority of incidents where your battleship becomes a landing craft.

There is another problem with CFS2 ships that is similar to sailing through land; they will also sail through each other without causing any damage. If you crash your plane into Martin Freeman the side of a ship you will be destroyed, but ships can pass through one another unharmed. There is no fix for this. The best way to avoid it is to keep your waypoints set as **Do not evade**. That will keep the ships on course. You are most likely to see this when a convoy with waypoints set to “Turn” is attacked and the ships go every which way.

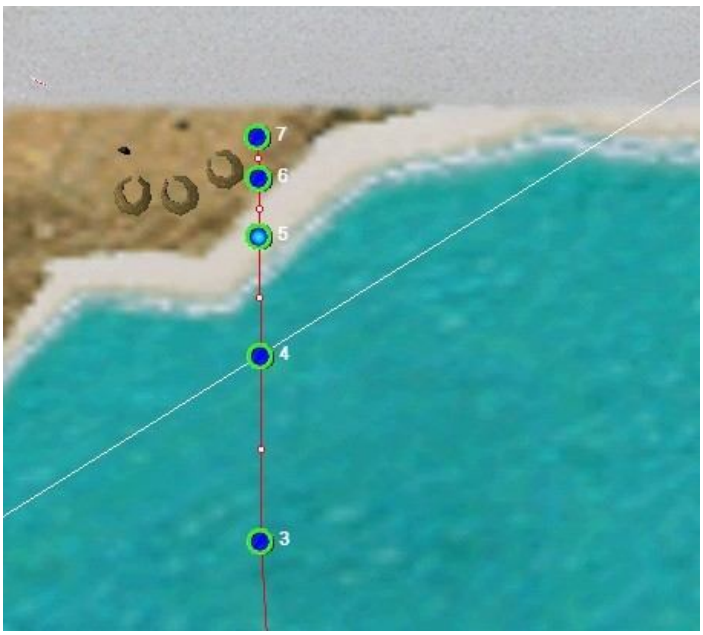
Stopping Landing Craft on the Beach

It's possible in CFS2 to simulate an amphibious landing by having your landing craft run up on the beach and stop. To make it work you need to follow a few simple steps and have an understanding of how ships follow waypoints.

A moving object gets its speed not from the waypoint where it is but from the **next** waypoint. So, if you plot a course with waypoint #1 set at 10 knots and waypoint #2 set at 30 knots, when the mission starts your ship will accelerate to 30 knots, not 10 knots. When a ship reaches its last waypoint, it turns back toward waypoint #1 and continues on at whatever speed waypoint #1 was set to. With that in mind, we can construct our beach landing.

Begin by plotting your landing craft's approach to the beach. I recommend that you not use a formation, but rather plot each landing craft individually. If you do a formation, all of the landing craft will stop when the primary one reaches the beach. That could mean some are well up on land and some are still in the ocean. Plot the first one then use the **Ctrl** key plus dragging to copy each additional landing craft.

I like to set my landing craft about a mile off shore and to use seven or eight waypoints to bring them into the beach. The trick is to start at a moderate speed and gradually reduce their speed until they reach the beach. In these illustrations you can see how I've plotted a landing on Tarawa. The speeds for each waypoint are **0** – 10 – 8 – 5 – 3 – 2 – **0**. Notice that the speeds at waypoint #1 and waypoint #7 are both set to zero. For this to work, both the first and last waypoints should be set to zero. You also need to step your speed down gradually as you near the beach. If you approach the beach too fast you will overshoot the landing zone and end up in the jungle. On the other hand, you note on the detailed illustration that the last waypoints are set some distance from the water. This is because a moving object responds to a waypoint shortly before it arrives there. You've seen this when flying in CFS2. We want our boats to keep moving until they reach the shore.



It takes some trial and error to get the speeds and waypoint locations correct, but if set up properly your landing craft will appear to run right up to the beach and stop, just like the real thing.

What happens next is another quirk of CFS2. You'll recall I mentioned that when a ship reaches its last waypoint it turns back toward waypoint #1 and proceeds there at whatever speed waypoint #1 is set at. Once your landing craft have reached the beach they may slowly pivot on their axis until they are pointed back at waypoint #1. Once turned around, since waypoint #1 has a speed of zero, they will remain stationary. It takes a long time for the landing craft to fully turn around and they may not do it at all, or only some of them may and some may not. Granted, it looks a little strange as they turn around, but flying over them at 200mph you are unlikely to notice the movement.

This technique works well for simulating amphibious landings and frankly, the turning around at the end, if it occurs at all, isn't noticeable.

☛ Some players have asked me if it's possible to get ships to engage each other in battle; they shoot at aircraft so why not shoot at other ships. So far that is not possible. Many people have tried (including yours truly) but no one has come up with a way to get ships to fire on one another. But hey, so what? This is a **FLIGHT** simulator after all, not a ship simulator. ☺

Ground Units (Vehicles)

CFS2 includes a variety of moving vehicles that can be added to your missions. To select a vehicle click the "**New ground unit**" button on the tool bar. This will place a truck shaped icon on your map. You plot the waypoints for your vehicles just like you do ships or planes and the same thirty-two waypoint limit applies. Your selection screen contains the same categories of 'Nationality', 'Unit ID', 'Skill', 'Goal', 'Start delay', 'Pre-damage', 'Hidden to player', and 'Invincible' that ships did. The Route Tab is also similar with one notable exception; the only action allowed at a vehicle waypoint is "Turn". Additionally the maximum speed for a ground unit is 13 knots.

Vehicles can be grouped into formations in exactly the same manner as ships, so you can create ground convoys to supplement your CAS missions as either friend or foe. There are just a couple of quirks with regards to vehicles that you should be aware of.

Unlike ships, vehicles will not "sail" through the land. They will follow the contour of the land and drive over the surface, climbing hills as they go along. Unfortunately, they treat water the same as land so if you aren't careful, your vehicles will drive across water the way ships sail across land. In this illustration you see a convoy sailing the North Atlantic on its way to England. Unfortunately the convoy is a convoy of Sherman tanks!

The solution is the same one you used with ships, only in reverse. Don't set waypoints too close to the water; avoid sharp turns, use multiple waypoints; slow down on curves. Plotted carefully, you can move your vehicles around the coastline without danger of turning them into little boats.

Vehicles follow other ship behavior as well. They will pass through each other without doing damage. Also, they always travel at ground level. That means that if you place an infrastructure bridge

over a river and route your truck convoy over the bridge, when they reach the bridge they will drive right through it at ground level, passing through the bridge as though it wasn't there.

Vehicle formations also turn the same way ship formations do; when the primary vehicle turns, all of the vehicles in the formation turn with it. This works for ship convoys and looks pretty realistic, but it looks out of place for vehicle convoys. Picture a real-life vehicle convoy driving along a road. When the lead vehicle gets to a corner it turns. Each subsequent vehicle keeps going straight until it reaches the corner and then it turns. In CFS2, when the lead vehicle reaches the corner and turns, all of the following vehicles turn at the same time. If you are making a right angle turn, your convoy will go from following in a single line to driving abreast.

Before the turn

After the turn

Depending on how close you set your vehicles and how sharp the turn, you can get some odd looking formations. If you make enough turns, you will find the vehicles all getting back in line again. While it would be nice to plot a convoy along one of CFS2's many winding dirt roads and strafe it with your Hellcat, the reality is that the end result may not be as pleasing to the eye as you had hoped.

Even with these limitations, moving objects add another dimension to CFS2. As you experiment with them and learn their capabilities you'll find new ways to incorporate them into your missions, adding realism and fun to the simulator. After all, watching a fuel truck and a couple of jeeps driving around the tarmac as you sit in the cockpit warming your engine really makes the sim come alive!

☛ Whether you're working with aircraft, ships, vehicles, or infrastructure, occasionally you end up with a lot of icons in the same place on the map and it can be difficult to select the one you want. An easy way to select the correct one is to place your cursor over the group and right-click. This will display a pop-up menu listing all of the objects at that location. Simply click on the one you want to edit and that object becomes highlighted.

Chapter ten – infrastructure

Preflight

Infrastructure is any non-moving object you can place into a mission. It includes buildings, tents, trees, anti-aircraft emplacements and guns, revetments, etc., along with non-moving ships, vehicles and planes. There are even infrastructure runways that you can place anywhere and use. Although we lightly touched on infrastructure in our missions *Raid on Tarawa* and *A "Wake" Up Call*, in this chapter we will take a closer look at them, how to use them, some of their quirks, and a few techniques for manipulating them.

Infrastructure Basics

We've already discussed selecting and placing infrastructure but it won't hurt to review that now. The "New infrastructure" button is located along the tool bar at the top of the Mission Builder screen.

When pressed, it brings up the Infrastructure tab where you select the object you want to place and can attach certain conditions. There are four classes of infrastructure: Aircraft, Infrastructure, Ships, and Vehicles. Under Aircraft you will see listed all of the planes in your AIRCRAFT folder, both player-flyable and AI. If you have added a player-drivable vehicle or ship to your AIRCRAFT folder that object will appear there as well. Notice that for the other three categories, Infrastructure, Ships, and Vehicles, the list starts with a group of objects preceded by an asterisk (*). These are old objects from the original *Combat Flight Simulator WWII Europe Series*, or as it is more commonly known, CFS1. You'll note that many of the objects are Europe specific. You can still use them in your Pacific missions the same as any other infrastructure object. Note however that they lack the detail of the newer CFS2 objects.

Infrastructure tab

CFS1 objects listed with an asterisk *

Next to the main Infrastructure tab is an Information tab. Click on this to bring up an image of the object and a brief description of what it is. This can be very helpful when trying to select the correct building or other object for which there are several variations. There are no pictures for CFS1 objects.

Information tab showing picture

Once selected, you can set the orientation of the object (more on this later). The 'Goal', 'Pre-damage', 'Invincible', and 'Hidden to player' options are the same as you've encountered elsewhere in the Mission Builder. Some objects, notably the stock aircraft, will show damage if the Pre-damage is set to 100%. It isn't perfect, but it does allow you to simulate wrecked planes scattered around

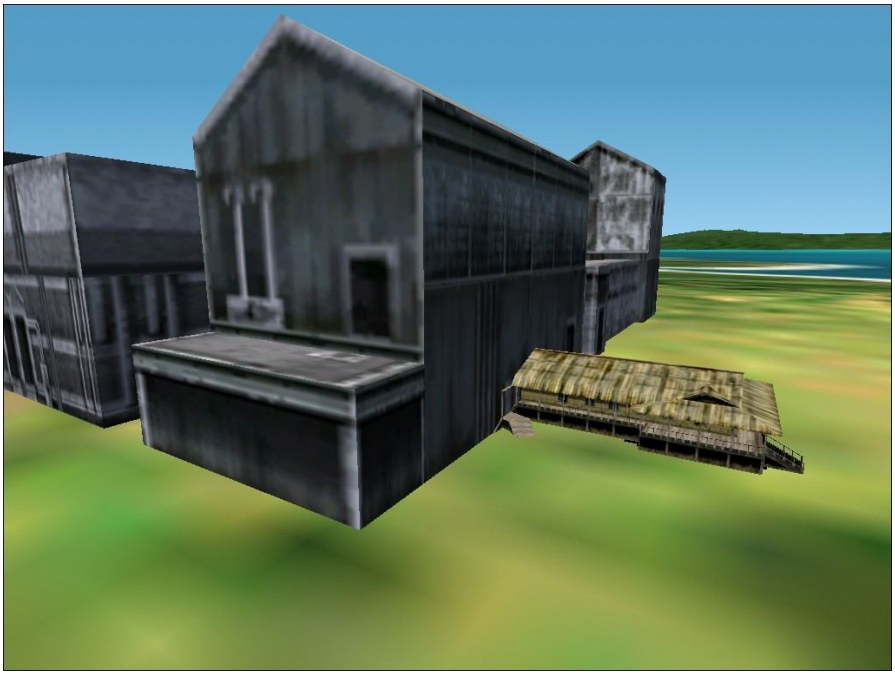
Damaged Allied aircraft

☛ You can add additional infrastructure to your CFS2 installation just like you do extra aircraft and ships. Several designers out there specialize in upgrading the stock CFS2 objects as well as creating all new infrastructure. Whether you are fighting in the Pacific or Europe, in summer or winter, in the jungle or in the desert, someone out there has designed new objects suited to your situation. Many of these far and away exceed the detail of the CFS2 originals. You owe it to yourself to visit the various flight simulator 'warehouse' sites and search for CFS2 downloads. Be sure to follow the installation instructions carefully. And, if you replace any stock infrastructure, be sure to make a backup copy of the original first. ☺

Unfortunately, infrastructure objects suffer some of the same problems as ships, along with a couple of others. Whereas ships will "sail" over land, infrastructure objects float on water. Infrastructure treats land and water the same so objects accidentally placed on the water will sit there as shown in the illustration on the next page of a "floating" base.

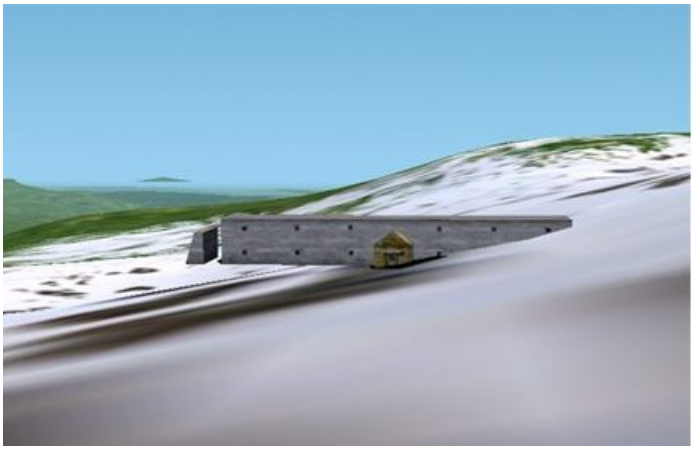
Floating Combat Base

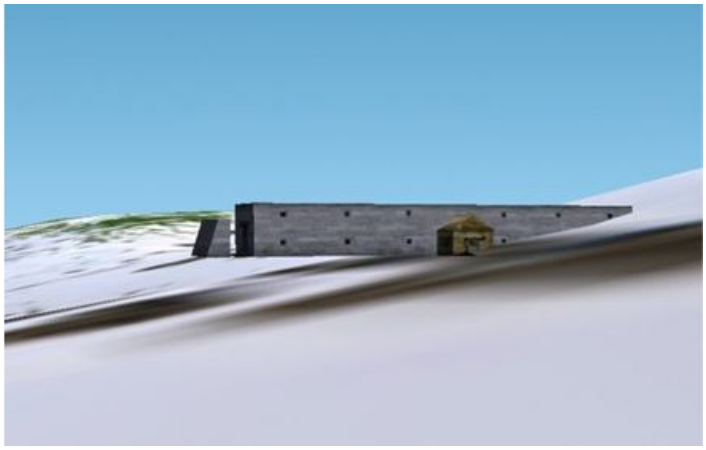
Like ships, infrastructure objects can collide with each other without doing any damage as seen in this illustration of a barracks stuck into the side of a factory. Actually, this isn't all bad. If you're creative, you might be able to design some clever looking structures by combining several buildings together.



Also, all infrastructure sits at ground level, meaning you can't stack infrastructure objects on top of one another. That means you can't place infrastructure planes on the deck of a carrier. They will end up sitting on the water beneath the ship.

There is one final quirk about infrastructure to be aware of. Infrastructure objects always sit parallel to the surface of the planet, not to the underlying terrain. You don't notice this when placing objects on level ground or ground that is only slightly angled. However, if you try to place an infrastructure object on a steep hillside you will see results similar to the next two illustrations. Notice that the radio building remains level but since it is on a mountain side, half of the building is stuck in the mountain and half is suspended in mid air.

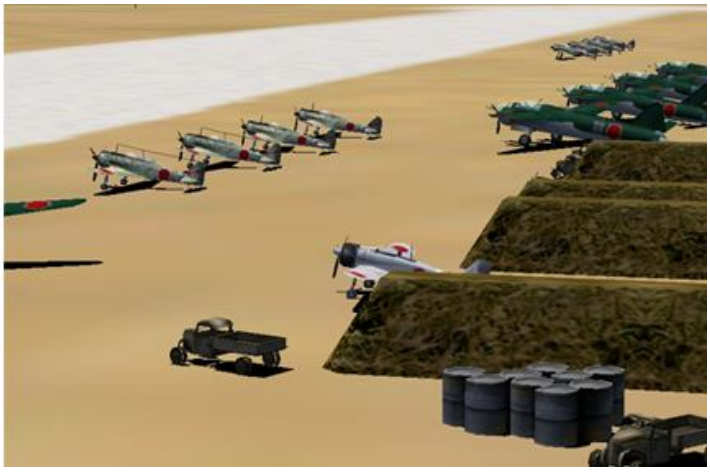




None of these issues are serious problems and once you understand the behavior of infrastructure you can compensate for any quirks in how you design your mission. For example, turn this radio building ninety degrees and it would look like a building with the back portion built against the side of the mountain; not perfect, but much more realistic than what we see here.

Placing, Grouping and Copying Objects

We already covered placing infrastructure in our missions *Raid on Tarawa* and *A "Wake" Up Call*; simply select the object you want, set the orientation, and click the spot on the map where you want it to go. CFS2 includes a limited number of infrastructure objects at each airfield. These are neutral objects, not aligned with any nation, so you can easily use them for either the Allies or the Japanese. They are the same objects that you place using Mission Builder. The only difference is that they are a part of the Global Scenery Layer (GSL). This is a semi-permanent layer of scenery that is called up by the simulator each time it starts. These objects can be destroyed and if you run into one you will crash. As seen below in these two versions of Yap Island, you can add additional infrastructure to an existing airbase to make it come alive. Remember however, that infrastructure placed into a mission using the Mission Builder will only show up in that mission.



Detail of improved Yap flight line

Manipulating GSL

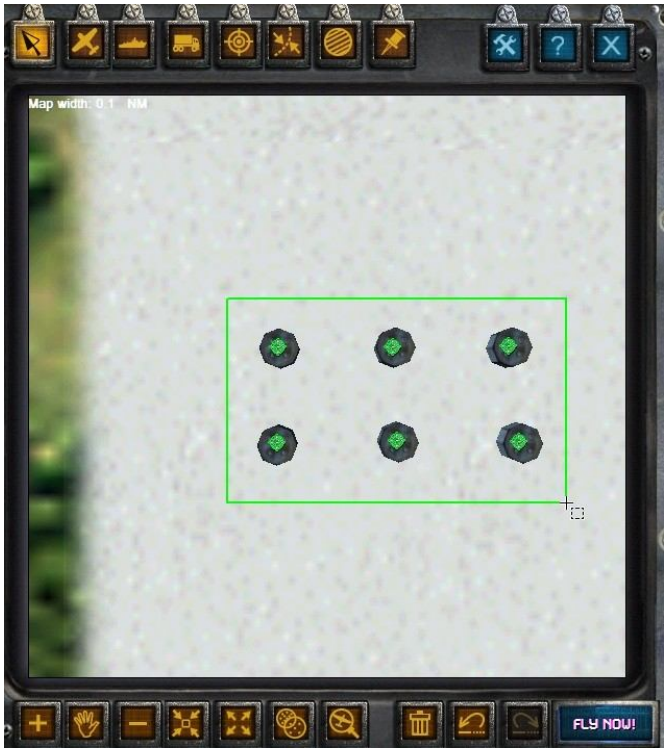
It's possible to add additional infrastructure directly to the GSL layer within CFS2. Once done, the new infrastructure becomes a permanent part of the scenery just like the stock infrastructure that appears at each airfield. This is not an exercise for the faint of heart. One mistake and you could wipe out all of the existing GSL scenery leaving your CFS2 stark and barren. However, if you are experienced with CFS2 and want to explore this technique there is some excellent information available out there.

Martin Wright (http://www.mnwright.btinternet.co.uk/index.htm) has created a GSL Editor that will allow you to add and remove GSL scenery objects from CFS2. It is not available for general download but he will provide it upon request if you email him. Read his excellent tutorial, **GSL Scenery Design for Combat Flight Simulator 2,** which comes with it before attempting any modifications.

Lindsay Watt (leroy10) has uncovered a way to add infrastructure to the GSL using the Mission Builder. This method is quick and easy but lacks some of the features of Martin Wright's GSL Editor. Check out Lindsay's GSL tips at his website (http://forum.lwdesigns.com.au/). You'll find his GSL instructions and all of his other fine add-ons in the Downloads section. Just remember – if you screw up your GSL, I warned you! ☺

When moving large groups of Infrastructure around it is much easier to move them as a group rather than individually. Simply place the cursor at one corner of the group, hold down the **Shift** key and drag the cursor to form a box around the group. The individual object icons will become highlighted. To move the group, drag any one of the icons to the new location and the group will follow.

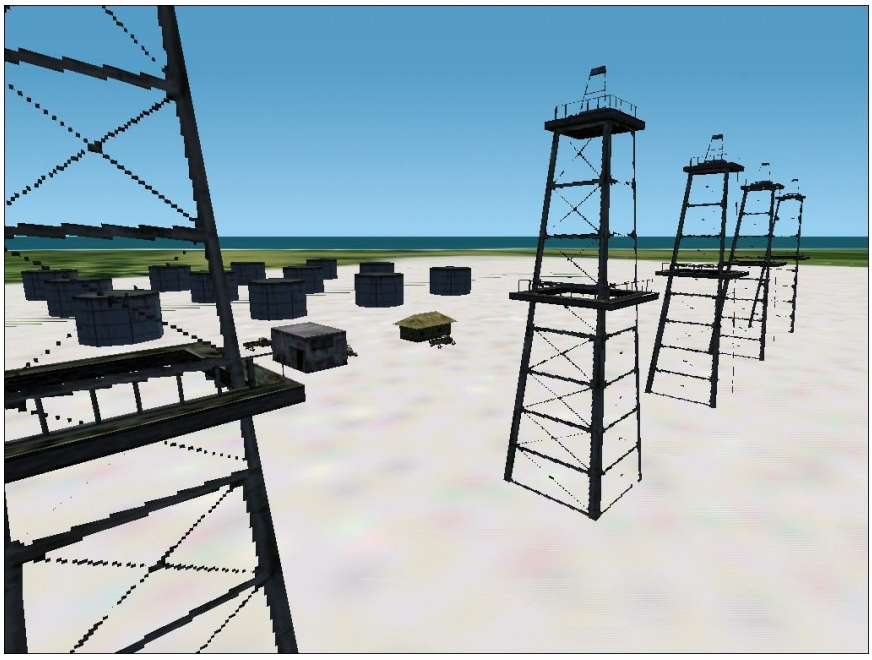
Similarly, you can quickly copy infrastructure objects to increase their numbers without having to select a new object each time. For example, say you wanted to create an oil field with lots of oil rigs and storage tanks. While you could place each object individually, that would take a lot of time away from flying the mission. Begin by placing five or six oil tanks where you want them. Then select the tanks by boxing them as described above. Finally, hold down the **Ctrl** key and drag any oil tank to copy the entire group. You can continue to select and copy in this manner until you get the amount you want. This is the same method described earlier in the Handbook for copying a flight of planes.



First set of oil tanks in place, boxed, ready to copy

Oil tanks completed, copying oil rigs

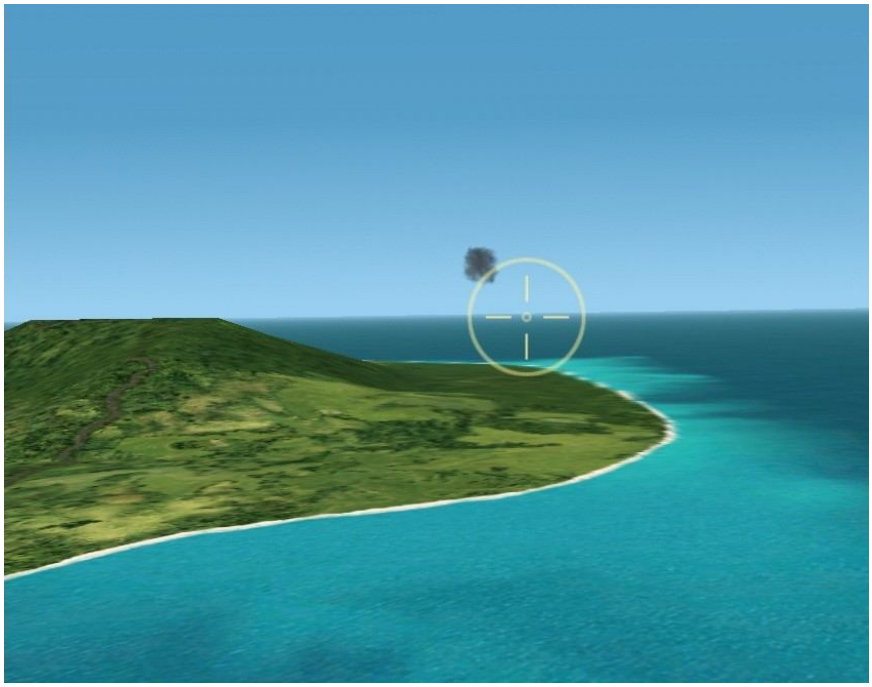
Here's a shot of the finished oil depot complete with buildings and vehicles ready to be bombed by my carrier based planes. The oil rigs are CFS2 objects while the oil tanks are some of the CFS1 objects included in the game.



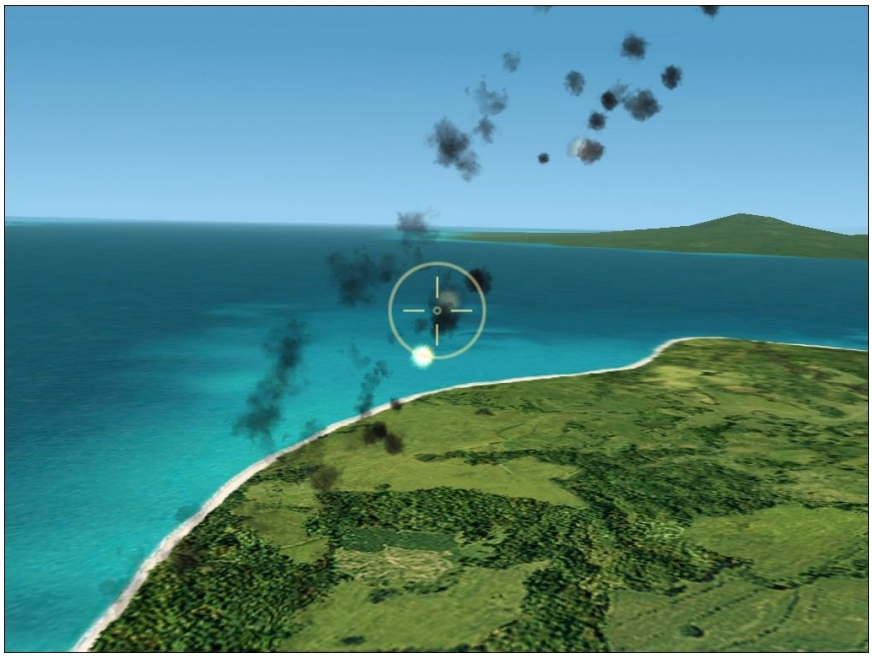
☛ By the way, I created the ground texture for the oil depot by using infrastructure coral runways set side by side

Orientation

Orientation is the direction an object is facing. You'll recall from Chapter Three that the default orientation is zero degrees or north. While you could place every object at the default orientation, your missions will look more realistic if you place objects so that they look more natural and face different directions. For some objects, such as trees, oil storage tanks, etc. orientation doesn't matter because the object pretty much looks the same regardless of which way it's facing. For other objects though, such as buildings, ships, and planes, orientation can make the difference between sharp looking scenery and scenery that doesn't make sense. Additionally, orientation can affect how a mission plays out.



Approaching the AA guns from the rear



Approaching the AA guns from the front

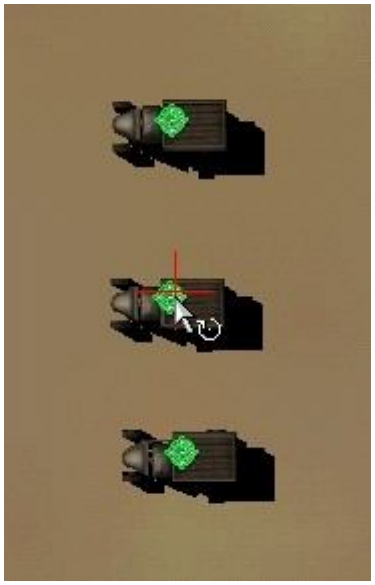
It would be easy to place all of your anti-aircraft guns at zero degrees to save time when building a mission. But experiments conducted by myself and others have shown that the direction an AA gun is pointed affects how it fires on the enemy. Look at these two illustrations. Fifteen AA guns of various types are in a line all pointed north, the default orientation. In the first illustration the player is approaching from due south, coming in behind the guns. In the second illustration the player is approaching from the north, directly into the guns. Notice the difference in the flak bursts.

In test after test, at various altitudes, the results were always the same, approaching the AA guns from the front resulted in heavier fire. There is no doubt in my mind that for the best results you should always point your AA guns outward, toward the enemy.

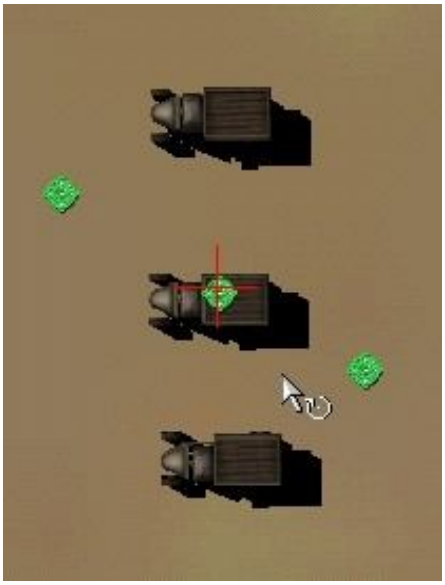
There is some variation in how the different guns behave and some will shoot at you when approaching from the rear; but for the best results align your AA guns the way you would in the real world, away from your base, guarding the approaches. It requires a little more work to orient the AA guns properly but you will see the results in the game.

Fortunately CFS2 allows you to group objects together and change the orientation of all of them simultaneously. With this technique you could construct one anti-aircraft gun emplacement with an AA gun in the center, then copy it to the different locations around your base and quickly change the orientation of each emplacement individually to face the proper direction without having to type in the orientation for each object.

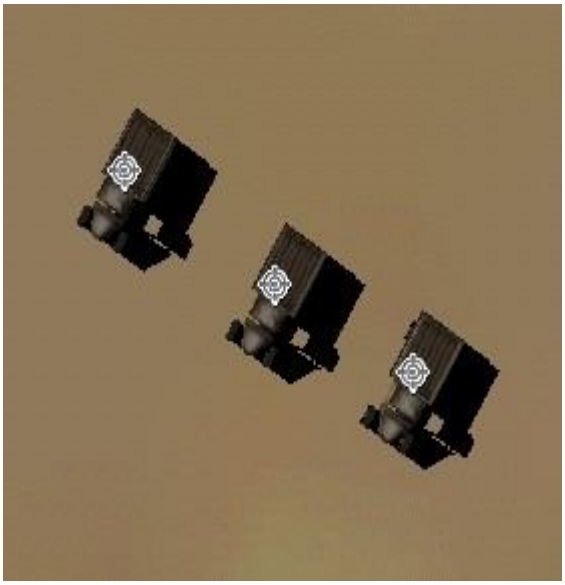
In this example I have three light trucks lined up in a row. To change the orientation of all three simultaneously begin by boxing them and selecting them as described above. While holding down the **Alt** key left-click directly on top of the middle truck. This will place a red cross where you clicked. This cross marks the rotation point around which the objects will turn. You will also see an icon that looks like an arrow formed into a circle. While still holding down the **Alt** key**,** place your cursor on top of any of the truck icons and left-click a second time. While holding down both the mouse key and the **Alt** key, slowly drag your cursor around in an arc. As you do so you will note that the icons for all three trucks will turn. When the icons are in the new position you want, simply release both the left mouse button and the **Alt** key. Left-click anywhere on the map to lock in the changes and you will see the images of the trucks jump to their new position.



Trucks selected – ALT pressed



Rotating the icons

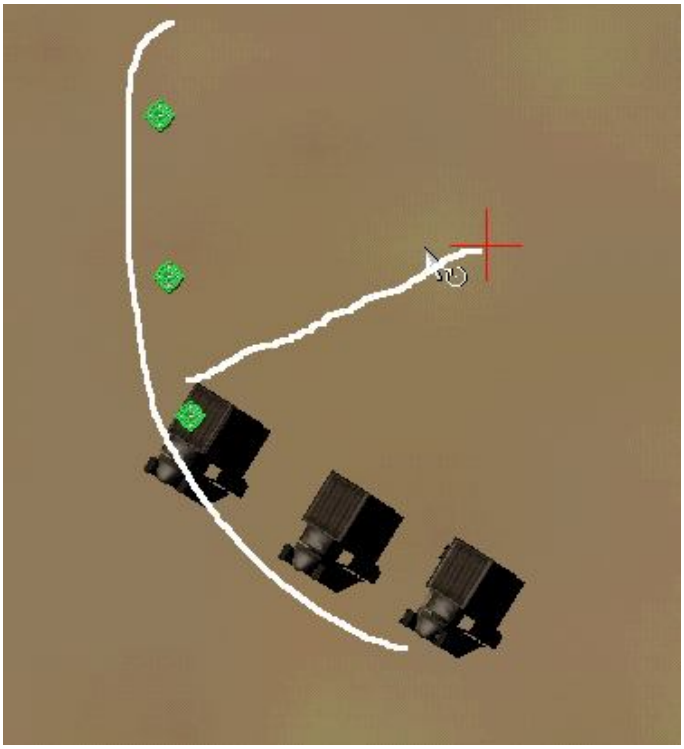


Trucks in their new positions

Remember that sequence for orienting a group:

1. Box the objects to highlight them
2. Place you cursor in the center of the group and press ALT
3. Continue to press ALT and left-click to create the rotation point (red cross)
4. While still holding the ALT key down, place the cursor on any object icon and rotate to the new position.
5. To lock in the new position, release the ALT key and left-click anywhere on the map.

Notice that the three trucks are still in alignment with one another but are now rotated about 45° to their left. What's nice about this method is that the objects keep their relationship to one another so if you placed an AA gun pointing toward the front of the emplacement, after rotation the gun is still pointing toward the front of the emplacement but both are now facing a new direction. The key is that once you have pressed the **Alt** key, don't let up until you have finished the rotation and are ready to lock the objects into their new position.



Where you left-click to place the red cross determines the point around which the objects will rotate. In the above example we used the center of the group of objects but you could place the cross father away if necessary. In this illustration note that I placed the rotation point above and slightly to the right of the center of the formation. The straight white line represents the radius of the rotation circle and the curved white line represents the path the trucks will follow when I rotate them. The farther away you set the red cross the greater the radius and the larger the arc the objects will follow.

Why would you want to do this? Well, maybe you want to move a group of objects from the top of the mission map to the bottom. For example, say you had a fleet of infrastructure ships positioned north of Wake and pointed away from the island. You then decided that you'd rather have the fleet south of Wake but still pointed away from the island. You could select them all and drag them to the south side of the island but they would be pointed toward Wake and you want them pointed away from Wake. Instead, use the group orientation feature, place your rotation point in the center of Wake and rotate them around the entire island. When you are finished they will have kept their relationship to one another and will now be pointing south as you intended.

Original position of fleet

Midway through rotation

Final position of fleet

If you look closely you'll see the rotation point near the center of Wake Island. Also note that in the last illustration I haven't yet done the final left-click. As a result you can still see the ship images in their original position even though their icons show the new position.

Try using this Mission Builder feature the next time you drop infrastructure onto the map. After you've done it once or twice it will make sense to you and you may find it a major time saver in the right situations. You can use this technique to rotate a single object as well, although you will probably find it much faster and easier to simply drag the object to the new location and enter the appropriate degrees of orientation into the window.

Layouts

Mission Builder allows you to save your favorite or most used collections of infrastructure objects as a layout which can then be used again and again in other missions. This can be particularly useful when writing campaign missions where you fly out of the same base over and over. If you have added additional objects to your base to dress it up, you can save those objects as a layout then reuse them for each additional mission. It is much quicker to add a layout than to replace each individual piece of infrastructure for every mission.

To create a layout, begin by placing all of the infrastructure objects where you want them. Then draw a box around all of the objects to select them as shown in the illustration below. With the objects selected, press **Ctrl+W**. That will bring up the **Save Layout** window. Give the layout a name and click **OK**.

Layout of a generic port ready to be saved

☛ Mission Builder includes infrastructure runways just like the ones at all of the stock airbases. You can place these runways into your missions just like any other stationary object and you can include them in your layouts. While you can land on one of these runways during a mission, you can't take off from one without modifying the mission file by hand. We'll cover how to do that in Chapter Fifteen of the Handbook. Due to a glitch in early versions of CFS2 these infrastructure runways may not show up in your missions even though you placed them correctly. If that happens to you, open your main CFS2 folder and follow these steps:

1. Copy the file **runways.bgl** that you will find in **SCENEDB\runways\scenery** and paste it into **SCENEDB\inf\scenery**.
2. Then copy the runway texture files from **SCENEDB\runway\texture** and paste them in **SCENEDB\inf\texture** as well.

To use the layout you created, press **Ctrl+L, t**his brings up the **Load Layout** window. Notice that you have the option of checking "**Place layout at original world location**". If you check this box your layout will be placed in the same spot where you first created it with all of the objects pointing as you placed them. This is the option you would select if you wanted to keep the scenery looking the same for each mission.

If you choose to leave this box unchecked you can place the layout anywhere in the world. This is useful when you have designed a generic base, dock, village, etc. and want to use the same basic design over and over without having to recreate it.

Instead of seeing the object icons you will see a cursor surrounded by small white boxes. Place this cursor where you want the layout to go then left click and the objects will appear. If they are not exactly where you want them, use the group orienting technique we covered above to rotate them into the proper position. Tweak any individual objects and you're ready to go.

Layout of generic port placed at new location

Final position after rotation and tweaking

Summary

Infrastructure objects in Mission Builder can make your sceneries seem more real. There are a wealth of add-ons available out there including new buildings, weapons, vehicles, and even people. Although you mostly fly over the CFS2 world at a couple of hundred miles per hour, it's still fun to make the environment look as realistic as possible when you are taxiing along the runway on your way back from a mission. The techniques you learned in this chapter will help you to get the most out of placing and using those infrastructures.

If there is a limit to the total number of infrastructure objects you can place in a mission I haven't found it yet. I have put hundreds of objects into my test missions without a crash. However, a large number of objects will affect your performance and you are liable to see your system slow down before you reach any maximum limit.

If you do start to experience slow downs there are a couple of things you can try. For starters, take a good look at all of those objects, maybe some of them can be removed without hurting the overall look and feel that you are attempting to achieve. Secondly, if you are always placing the same layout in the same location you might want to consider adding those objects to the GSL layer as described in an earlier sidebar. GSL objects do not reduce frame rates as much as infrastructure objects. If you do this however, remember that those objects will <u>always</u> appear at that location. Take that into consideration when placing objects that are specific to one nationality or another. GSL is still not a cure-all. Too many GSL objects will also have a negative impact on system performance, but it will take many more objects before you start to notice slowing.

Chapter eleven – Front Lines

Preflight

Front Lines are used to differentiate between friendly territory and enemy territory. They are not essential, but they can be very helpful and you may want to get into the habit of using them. Additionally, front lines can have an unexpected impact on triggers, events and mission goals. In this chapter I'll show you how to use front lines and explain how their use can help or hurt your chances of completing a mission successfully.

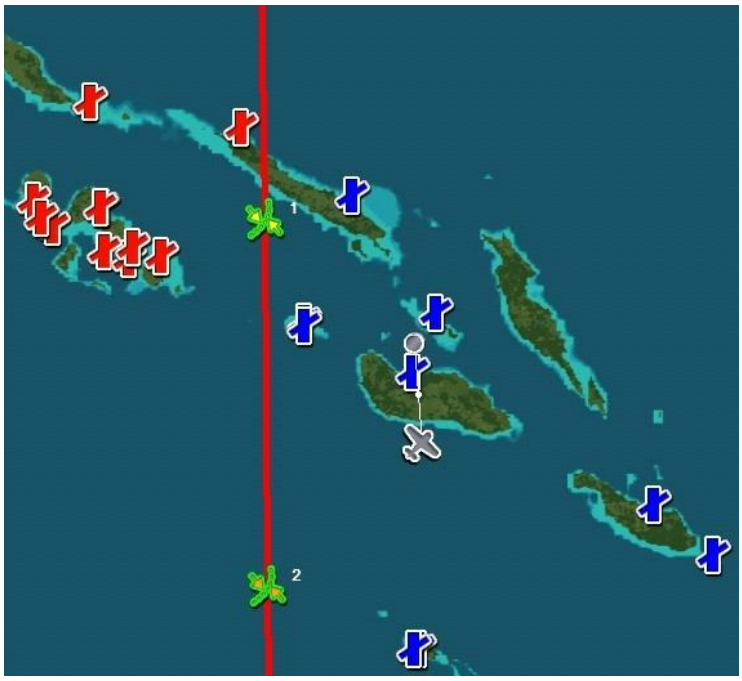
Using Front Lines

To place a front line, begin by clicking the **New front line** button on the tool bar. It looks like two arrows pointed at each other separated by a dotted line. You need a minimum of two points to establish a front line. Pick a point anywhere on the map along the front line and click. Then pick a second point also along the line and click again; right-click to finish.

You have created a straight front line that now separates friend from foe. **Everything on your side of the line (the side where your aircraft icon is located) is friendly territory; everything on the other side of the line is enemy territory.** Although you can only see a small portion on the map, your front line extends all the way around the world. No matter how far up or down the line you fly, you remain in friendly territory until you cross over to the other side. You probably noticed that when you created your front line the colors of any airbases shown on the map changed.

All of the default airbases in CFS2 are neutral; they are not aligned with any nationality. All neutral bases show up as grey "X's" on the map. When you place a front line, all of the friendly bases turn into blue "X's" and all of the enemy bases turn into red "X's" as shown in the illustrations below. This helps you to keep track of which is which in the Mission Builder as well as on the Advanced Info map.

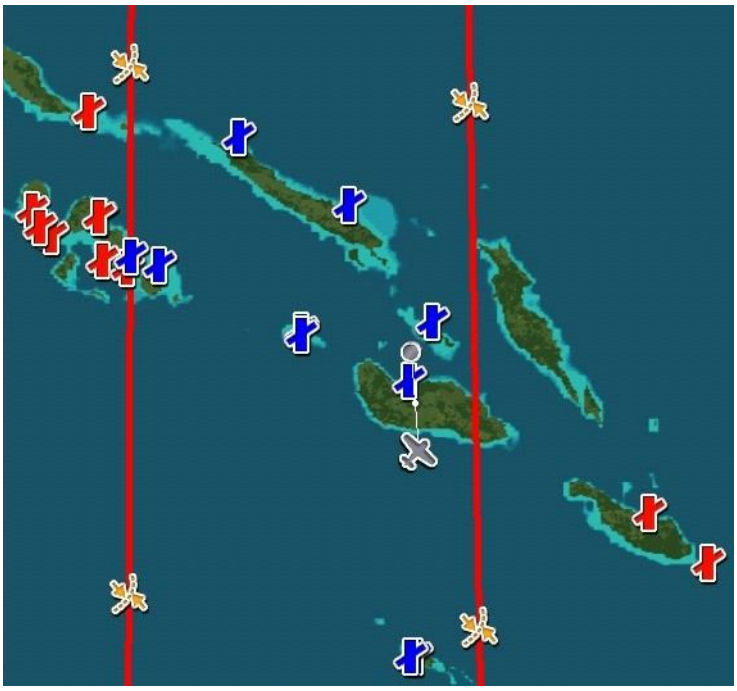
No front lines, all bases neutral



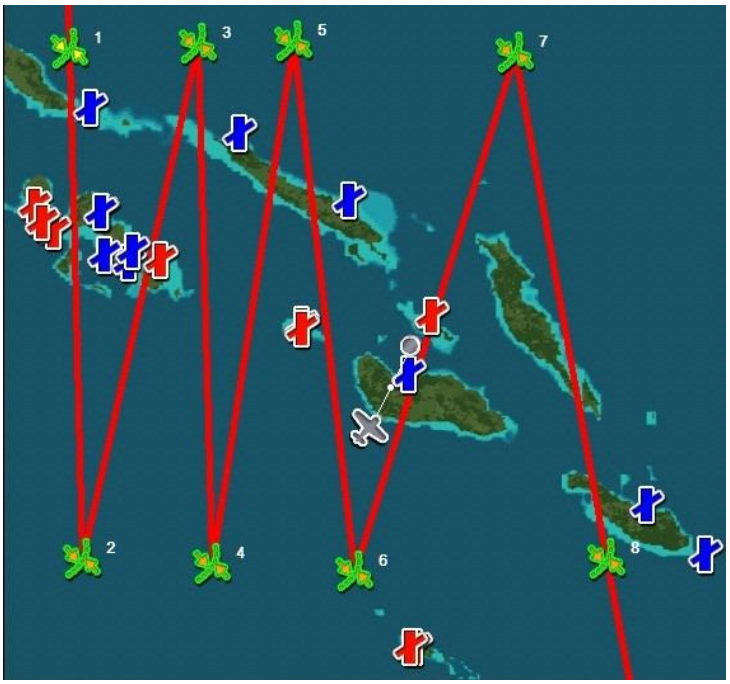
Enemy bases in red, friendly in blue

If you need to tweak the position of your front line simply click on one of the front line icons and drag the line to reposition it.

Mission Builder allows you to use up to two sets of front lines so you can create zones where the player flies in and out of friendly territory. You can also add additional front line segments by right clicking on any front line icon and selecting **Insert**. By adding segments you can draw your front lines into different patterns, both realistic and absurd.



Multiple front lines using two sets



Multiple front lines using a single set

Two friendly islands surrounded by enemy territory

Iwo Jima designated as enemy territory

Where's the "X"?

In the illustration of Iwo Jima you can see three runways but no "X"'s to mark them. What gives? Only runways entered into the **airbases.dat** file will show up in Mission Builder with an "X". And only airbases with an "X" can be designated **Take off** or **Land at base**. All other airbases will be visible in Mission Builder as long as their scenery is activated, but you may have to zoom in close to see them. We'll cover the airbases.dat file in Chapter Eighteen

Front Lines and their influence on Infrastructure, Triggers, Events, and Mission Goals
Aside from simply designating friendly and hostile territory, front lines can affect the outcome of a mission or campaign. If you bail out over friendly territory you will be picked up by a submarine and survive to fight again. In a single mission this isn't all that important, but in a campaign it keeps your pilot alive and still flying. However, if you are forced to bail out over enemy territory you will be captured. In a campaign, this ends the campaign right there – you're finished! Microsoft uses front lines in the stock campaign missions and you should too in any campaigns you design. Front lines (and the chance of being captured) add both realism and greater difficulty to a campaign mission and should be considered a basic element of your mission design.

A lesser known effect of front lines is the way they interact with infrastructure. Like ships and planes, each infrastructure object has a DP file. Aside from containing the information regarding how the object can be destroyed and any guns it is carrying, the DP file also designates the objects nationality. Infrastructure objects can be designated Allied (American or British), Axis (Japanese or German) or neutral (non-aligned). Some objects such as tanks and AA guns are clearly aligned with one side or another but many of the objects in CFS2 are not associated with any nationality. Things like crates, revetments, tents, buildings, oil cans, etc. are neutral and can be used by any nation. Unfortunately there is a glitch with the allegiance of the old CFS1 objects. The British vehicles are aligned with Japan and the German vehicles are aligned with the USA. The buildings are neutral.

You may recall that one of your parameter choices under the trigger type "Status" is "Ground targets destroyed". You can use this trigger to trigger an event. You can also make the event of destroying a certain number of ground targets a mission goal. Here's the catch: <u>Only Enemy Ground Targets Destroyed Count Toward the Total.</u> You can destroy a hundred tents and never reach your goal of destroying four or more ground targets as long as the tents remain neutral. To get neutral infrastructure objects to count toward your goal you must change their allegiance first.

There are two ways to change the allegiance of an object. One is to modify the object's DP file and delete the CDP file. This would be a very time-consuming way to do things and it wouldn't work well because changing the allegiance would mess up all prior missions created using that object. The other way and the best way is to use Front Lines.

By using front lines, every infrastructure object on your side on the line will be designated "friendly" and all infrastructure on the other side of the line will be "enemy". This is an across the board change of allegiance so it applies to objects such as palm trees as well. That means destroying a palm tree carries as much weight toward meeting your goal as destroying a factory.

Using front lines in your missions, especially campaign missions, is one more method of keeping your missions interesting and challenging. Try it!

☛ There's a quick way to tell if you are destroying friend or foe infrastructure in a mission – look at the status message. When you destroy a friendly or neutral object the message appears in **RED** text. When you destroy an enemy object the text appears in **WHITE** text.

Chapter twelve – sounds

Preflight

From the whine of your engine at start up to the chatter of your machine guns and the squeal of your tires on touch down, sounds add an air of realism and authenticity to CFS2. The sounds in CFS2 come from a variety of locations within the game. Some of these sounds are called up automatically by the game. Some are readily available for use in the Mission Builder and can be added as a part of an event. Plus, you can download or create your own custom sounds and add them to the game for use in missions as well. In this chapter we'll take a brief look at the sounds of CFS2, where they are and how to use them in your missions.

Sound Locations and Functions in CFS2

The sounds of CFS2 are found in the following locations:

- **Individual aircraft folders** – These are sounds associated with an individual aircraft and generally include the engine sounds and other sounds of flight but may contain other non-voice sounds as well, especially in add-on aircraft. Every aircraft either has its own sounds or borrows the sounds of another plane through the act of aliasing.
- ****\SOUND –** This folder contains basic game sounds such as guns, bombs, explosions, wind, the sound when your chute opens, etc., as well as all of the sounds associated with vehicles and ships. It also contains sounds left over from CFS1, some of which are not used in CFS2 such as the sound of the V1 Buzz Bomb. You should not delete any of these sounds, although some designers have produced different gun and explosion sounds which can replace the stock sounds if you wish.
- ****\SOUND\action** – This is the most important folder to the mission builder as this is the folder which contains all of the voice commands and comments that you can add into a mission event. These sounds include not just combat sounds but the entire range of instructor's commands from the training missions as well. The sounds are in both Japanese and English. The file name for each sound corresponds to the list seen in the event window. If you want to add custom sounds into your missions you must place those sounds into this folder for Mission Builder to list them when creating an event.
- ****\SOUND\PRECIP** – These are the sounds associated with rain.
- ****\SOUND\radio** – These are the sounds of radio transmissions generated automatically by CFS2. When you hear "*Radar reports bandits in your area*", this is where that sound comes from. If you look closely you'll note that each group of sounds has three files associated with it with the extensions **gvi**, **gvr**, **gvt**. The gvi files appear to be used by the game to control which sounds are called up and to insert the proper sound and text into the game. The gvr files are groups of sounds clustered together into a single file. Each gvr file may contain several dozen individual radio transmissions. The gvt file contains the text of the sound found in the matching gvr file. It is this text that you see on the screen when you hear the radio transmission over your speakers.
- ****\SOUND\THUNDER** – These are the different sounds of thunder associated with a thunderstorm.
- ****\SOUND\VO** – These are the narrative sounds heard during a campaign. At certain points in a campaign you will watch a short video and hear an explanation of what is happening in the war. This is where those narrations come from. The text file of the same name in the folder is the text that appears on the screen while the video plays. You should not delete these files. However, you can create you own video and sound files and incorporate them into your campaigns. Refer to Chapter Seventeen for more information.

☛ If you're interested in exploring what you can do with sounds, you need to pick up a copy of Martin Wright's utility **SplitGVR**. This great little tool allows you to open any of the **gvr** files and extract the individual sounds. You can then save them into your **\SOUND\action** folder and insert them into your missions so that they play when you want them to. It also allows you to replace an existing **gvr** sound with a new one. This is a must have utility for anyone interested in CFS2 sounds – and it's free! Just go to Martin's site: http://fly.to/mwgfx/ and click on "Beta Programs". Scroll down until you find **SplitGVR**.

Using Sounds in Mission Builder

To add a sound to a mission you must enter it in as part of an event. Under the action type, scroll down and click on either "**Play priority sound**" or "**Play sound**". Any sound in a mission can be interrupted by another sound. If you select "Play priority sound" the sound you chose won't be interrupted.

When you click on either "Play priority sound" or Play sound" it will bring up a list of 563 sounds (radio transmissions) that you can choose from to add to your mission. These sounds are listed by their meaningless file name but you can click "**Test**" to hear the sound. Simply select the sound you want to add and click the right arrows (**>>**) to add the sound to the mission. For example, you could create an event where the trigger is a carrier being sunk, then select sound **w640** as the action. When the carrier sinks you will hear the radio transmission: "**Scratch one flattop!**" To assist you in finding the best sound, I have created a transcript of all of the English language sounds in the stock game. They can be found listed in **Appendix 'B'** at the back of this Handbook.

Don't be trapped by using just the stock sounds. You can add your own radio transmissions and other sounds as well. Any **wav** sound can be added to the \SOUND\action folder and used in a mission. I occasionally add music to my missions this way. Here are some examples of how I use custom sounds and music:

- When the mission starts my RAF player is sitting in the cockpit when he gets word to stand down, the war is over. I instruct the player to go to Spot View and pan out. Then I set off a fireworks display (using add-on colored flares) while playing *God Save the King*.

- On a long B-17 flight where much of it is flown in real time I've added over an hour of Big Band hits from the war years. It is much the same as tuning in a distant radio station and flying the beam.

- Your base is under attack as the mission starts and you're sitting in the cockpit. You can hear air raid sirens and bombs going off in the background as you start your engine and get airborne. A variation of this, flown off a carrier, includes sounds of a klaxon and the call to General Quarters coming over the PA system.

- You're part of a large air attack against an enemy fleet, lots of ships, and lots of air to air combat. The radio is full of chatter as pilots call out about targets hit, bandits approaching, friendlies in trouble, etc. See the sidebar at the end of this chapter on how to easily set this up.

If you add music, or other long transmissions, and want to play several back to back you must build in a sufficiently long delay to cover the length of the song. In other words, if I want to play two songs back to back and the first song takes three minutes (180 seconds), I would enter the song, then enter a delay of 188 seconds (the length of the song plus 8 seconds of dead air) followed by the second

song. If you don't leave sufficient time for the first song to complete you will not hear the second song.

You can also record and use your own sounds and radio transmissions. While you can use the built-in recorder that comes with Windows, I'd recommend getting something more sophisticated if you plan to get serious about recording your own sounds. There are many programs out there but my personal favorite is **Audacity**. You can import, export, record, trim, edit, etc. with just a few mouse clicks. It is easy to use and intuitive to learn. Best of all, it's free! You can find Audacity here: http://audacity.sourceforge.net/.

Sounds in the European Theater of Operations

Although there are some CFS1 sounds included in the \SOUND folder I don't know that CFS2 actually uses them and none of them are radio conversations. And, since CFS2 is set in the Pacific, all of the radio chatter is in American English or Japanese. That isn't a problem for the stock game or any missions set in the PTO. But some players have a second, European Theater, installation and many of them are looking for British and German radio traffic.

British and German language radio transmissions are available for download as add-ons from Simviation (www.simviation.com) as well as other places. These are wav files and can be dropped into your \SOUNDS\action folder and inserted into missions as described above. They work very well. The problem is that you need to create triggers and events to add them; they are not called up automatically by the computer. In fact, when flying for Britain you will hear the same, computer generated, American accented chatter you hear in the Pacific. For many ETO enthusiasts, that isn't good enough.

Thanks to Martin Wright's SplitGVR, described in an earlier sidebar, a solution may be in sight. SplitGVR allows you to replace the canned radio transmissions with different ones. With a little bit of effort it would be possible to replace the American and Japanese radio chatter with British and German. My understanding is that several designers are attempting to do just this and by the time you read this, new sounds may be available for installation in your European Theater version of CFS2. Check the major flight simulator warehouse sites and their forums for the latest on this project.

Summary

Sounds are one more element in the design of a complete and immersive mission. Familiarize yourself with the stock sounds in CFS2, incorporate them into your missions, and don't hesitate to try importing your own custom sounds to dress up your missions and make them feel more realistic.

☛ Here's a tip for adding lots of sounds without having to create a new event each time. In the example mentioned above I have the airwaves full of radio chatter during an attack on an enemy fleet. To set this up I created one Trigger/Event for arriving at the battle site then I added all of the sounds I wanted to use to this single event, separating each sound by delays of differing lengths. The whole thing plays out over about 10 to 15 minutes. By adding lots of radio chatter and varying the times between each, it sounds very realistic. The sky is full of aircraft and flak and the radio is full of excited voices. With so much action in the sky you're never in the same spot twice, so each time you fly it's a little different. The background noise adds to the effect.

Part Three

enhanced Mission Building II

A few techniques for hand-editing your mission files to maximize your mission building experience.

Chapter thirteen – hand-editing basics

Preflight

By now you have a good handle on what the Mission Builder is capable of. Using the various tools available, you can construct exciting and challenging missions that are fun to play over and over again. If you so choose, you can design historically accurate missions that allow you to relive history. At the same time, you are probably beginning to understand Mission Builder's limitations. As good as the basic program is, there are some things that can't be accomplished within the framework of the interactive program.

Hand-editing your mission allows you to go beyond basic mission building and accomplish things the original program wasn't designed to do. For example, maybe you want to fly as a wingman, as shown in the illustration above. Note that the player's plane is number five in line, with the cockpit open and the engines still turned off. You can't do this within the Mission Builder program, but with a few simple edits you can place yourself in any position within the group, or even fly a different plane than the rest of the formation. In this section we will explore some of the edits that you can perform. This isn't a comprehensive list of everything that's possible, but it will get you started in the right direction.

The Basics of Hand-Editing Missions

You may recall from an earlier sidebar that all CFS2 missions will have either one or two files associated with them. All missions will have a file with the extension **mis**. This is the basic mission file itself. It contains the information on the aircraft, ships, vehicles, and infrastructure used as well as all waypoint information, weather, callsigns, nationalities, date, time, and the preflight briefing as well as some other information. The second mission file is the **dyn** file. This file contains all of the **dyn**amic information, in other words everything related to triggers and events. While the preflight briefing is contained in the **mis** file, the debriefing is contained in the **dyn** file. If your mission does not use triggers and events it will not have a **dyn** file associated with it. The names of the two files are the same, only the extension is different.

Both the **mis** and the **dyn** files are simple text files and can be edited with any text editor. I prefer **Notepad,** which comes with Windows, and will refer to it throughout these chapters, but you are free to use any program that you are comfortable with. If you plan on doing much hand-editing of your missions you may want to associate both the **mis** and **dyn** files with Notepad so that you can simply double-click to open the files.

To associate Notepad with either file, right-click on the file icon and select **Open**. On the pop-up screen click ***Select the program from a list*** then click **OK**. On the next screen be sure ***Always use the selected program to open this kind of file*** is selected. Then browse until you find Notepad, select it and press **OK.**

Once you've associated Notepad with the **mis** and **dyn** file types their icons will change to a Notepad icon. Now anytime you want to open a mission file you only need to double-click on it. Some simple things can be more quickly edited by hand than by opening up Mission Builder and doing it that way. In this chapter we'll look at the two mission files and point out those items that can be easily changed.

Open up both the **mis** and **dyn** files for our mission *Luck of the Draw*. You will note that while both files are text files they are very different in how they are laid out. Let's look first at the **mis** file.

```
[mission]
version=2.0
title=%title_string%
summary=%summary_string%
objective=%objective_string%
intelligence=%intelligence_string%
mission_type=8
allegiance=3
total_goals=1
player_aircraft=3842
player_aircraft_name=%player_aircraft_name_string
%
airfield=%airfield_string%
date=1/6/1943
time=12:00
```

<u>The MIS File</u> You'll notice a number of lines that look similar to **title=%title_string%**. All of these are pointers to somewhere else in the mission file. In plain English this entry says: *The title for this mission will be found in the title_string*. If you scroll down to the bottom of the mission you'll see a section called **[strings]** followed by the title string: **"title_string"=Luck of the Draw**. Similarly, there are strings for each of the other mission texts.

Allegiance "**3**" equals American. This is the nationality under which the mission will show, not the player's nationality. If you changed this to a "**1"** you would have to look under **British** to find your mission. This can be easily changed if needed.

The player aircraft number is a number assigned by the program to your aircraft for this mission file. This number will change from mission to mission and your copy of *Luck of the Draw* may have a different number. Knowing this number will help you to find references to your aircraft later in the mission file.

The date and time can be easily modified by hand.

If there were any custom weather it would be entered next in a section called **[weather]**. As we chose clear skies there is nothing showing in our mission. We will cover custom weather in Chapter Fourteen.

☛ After making your hand edits to a mission you must save the mission. If you want to overwrite your existing mission simply click on **File – Save**. To save your edited mission with a new name, click on **File – Save As**. Then give your mission a new file name. Since Notepad is a text editor it will try to save the mission as a text file. To save it as a **mis** or **dyn** file you must use the full file name, plus the extension, and enclose the whole thing in quotation marks like this: **"CC_HB_08.mis"**. Now your mission will save as a CFS2 **mis** file. Remember, both the **mis** and **dyn** files must have the same file name for them to work; only the extension is different.

```
[unit.0]
id=9156
type_id=3842
name=%name_string.0%
is_player_aircraft=1
allegiance=3
skill=2
aggression=2
payload=0

[unit.1]
id=9157
type_id=3842
name=%name_string.1%
allegiance=3
skill=2
aggression=2
payload=0

[formation.0]
id=6000
type=air
waypoint_path_id=5000
directive=8
players_formation=1
formation_type=1
points=2
unit_id_position.0=9156,0,0.0,0.0,0.0
unit_id_position.1=9157,0,-300.0,-600.0,0.0
```

The next sections contain information on the various flights. The player's aircraft is always listed first in his formation but could be any unit number and may not be the first formation listed. In this example he is **[unit.0]**. He is further identified as the player by this line: **is_player_aircraft=1**. Since we had one wingman, he is shown as **[unit.1]**. Additional wingmen would be numbered accordingly.

The allegiance field is the allegiance of the player and/or wingmen. If necessary you could change this. Although it wouldn't be noticeable in the game, you could fly as an American with an Australian wingman.

The skill and aggression levels are as follows:

Skill 1 = Rookie	
Skill 2 = Veteran	Aggression 1 = Average
Skill 3 = Ace	Aggression 2 = High

When the aggression is set to "Low" nothing appears in the mission file. You can edit these easily and may want to have different skill levels among your different wingmen, something you can't do from within Mission Builder. Obviously the entries for the player are meaningless. You determine your skill and aggression level, not the game.

payload=0 is the first payload listed in the plane's DP file, usually 'guns only'. **payload=1** would be the second payload listed and so on.

The units are followed by the formation information. The ID number here, 6000, is used by the **dyn** file. Formation type **1** is for Fingertip.

The key numbers in this section are found in the **unit_id_position** sections. Note the ID numbers in each section; they correspond with the ID numbers found in the Unit sections above. The player (9156 in this case) is always listed first. Next to your wingman you see two numbers in bold, **-300** and **-600**. These numbers represent the relative position of the wingman to you at the start of the mission. The first number is how far left or right of your position your wingman is and the second number is how far in front or behind you he is. Minus numbers indicate to the left or behind, positive numbers indicate to the right or in front. All measurements are in feet. Therefore, our entry of: 0**,-300**.0**,-600**.0,**0**.0 tells us that at the start of the mission our wingman will be sitting 300 feet to our left and 600 feet behind us.

The zero shown in bold is the relative altitude of our wingman to us at the start of the mission. Since we start by taking off from Henderson this number is **0**. Had we started in the air, it still would have been **0** because the first wingman starts level with us. However, had we had a second wingman his entry would have looked like this: **unit_id_position.2=9158,0,300.0,-600.0,150.0**. Our second wingman would be starting 300 feet to our right, 600 feet behind us and **150** feet **above** us. A negative 150 (-150) would have indicated he was 150 feet **below** us. All of these numbers can be edited to change the relative position of you and your wingmen to each other.

```
[waypoint_path.0]
id=5000
waypoint.0=2,1,S9* 25.87',E160* 2.85',+0,,,,0,,8030,0,1
waypoint.1=1,1,S9* 16.97',E160* 31.00',+3500,,,,219,,0,1,1
waypoint.2=1,1,S9* 12.21',E160* 14.70',+1000,,,,219,,0,1,0
waypoint.3=1,1,S9* 3.90',E160* 3.57',+1000,,,,219,,0,1,1
waypoint.4=1,1,S9* 3.99',E160* 1.45',+1000,,,,219,,0,0,0
waypoint.5=4,1,S9* 25.87',E160* 2.85',+0,,,,0,,8030,0,1
```

The next section contains information on your waypoints. We have six waypoints so there are six entries. However, in the mission file they start with waypoint #0 whereas in Mission Builder we started with waypoint #1.

Each waypoint entry contains the longitude and latitude of the waypoint along with your altitude and speed. Note that for waypoints #0 and #5 the altitude and speed are zero. That's because we set those waypoints as Take Off and Land at Base. The number **8030** is the ID number for the airbase at Henderson. This matches the entry in my **airbases.dat** file and in the runway information near the bottom of the mission file. Note also that the altitude at waypoint #1 (our #2 in MB) is **3500** feet and the speed is **219** knots, exactly what we entered in Mission Builder. You can easily change these settings through hand-editing if you like.

The mission file next contains similar entries for all other moving objects entered in the mission. Each moving object, or formation, will have a unit section, formation section, and waypoint section. Scroll down through the mission file and you'll see entries for the two flights of Zeros and the two Torpedo Boats. Each will be slightly different but all will follow the basic formats shown above. You can edit each of them in the manner that we discussed for the player's flight.

```
[unit_type.0]
type_id=3810
unit_family=1
name=A6M5_ZERO

[unit_type.1]
type_id=3842
unit_family=1
name=F4F4_WILDCAT

[unit_type.2]
type_id=6412
unit_family=5
name=shj_torpedo_boat
```

Below the last formation entry will be a list of entries showing all of the planes, ships, vehicles and infrastructure that you placed into the mission. Each has a **type_id** which matches the type_id shown in the respective unit section. The unit family information comes from the objects DP file and can be edited when necessary. For example, you may have noticed that the Japanese Betty bomber breaks formation and fights like a fighter. That's because the unit family is set as a **1**, **fighter**, instead of as a **2, bomber**. By changing the unit family in the mission file you would correct that problem. Of course, the best way to handle that particular fix is to change it in the Betty's DP file, then the change is permanent and will apply to all future missions using the Betty.

The most frequent thing you will probably change here is the name of the object. Say for example that you wanted to permanently change the torpedo boat to a Japanese troop ship, simply edit the torpedo boat entry to show the troopship's folder name (**SHJ_TROOP_SHIP**). From then on every time you fly the mission troopships would spawn instead of torpedo boats. You can change the player's aircraft from inside the game, either prior to starting the mission or anytime during the mission by clicking **Alt→Aircraft→Change/Modify Aircraft**. However, that will only change the player's aircraft, not the wingmen. To permanently change the player and wingmen's aircraft, to the Hellcat for example, edit the Wildcat entry to show the Hellcat's folder name: **F6F3_HELLCAT**. Now the mission will always start with the player and his wingmen in a Hellcat instead of a Wildcat. See the sidebar *What's in a Name* for more information on aircraft naming conventions within the **mis** file.

What's in a Name

You may notice that the names for the aircraft shown in the **mis** file are different than the names shown on the drop-down menu within Mission Builder. There are some differences in the naming conventions used within Mission Builder, Free Flight, and the **mis** file. Based on my experience, here are some general rules:

The aircraft names displayed in the Mission Builder (MB) drop-down menu come from the title line in the plane's **aircraft.cfg** file. If you have two planes with the same title, that name will appear twice in MB. That isn't true in Free Flight (FF).

Generally in Free Flight the names displayed are again from the title line in the **aircraft.cfg** file. However, in FF if two planes have the same title the game displays the title name for one of the planes and the folder name of the other.

Multiple skins (textures) can cause issues in MB. If you have a plane with multiple skins and different **aircraft.cfg** title entries for each skin, you will see the title for each plane in the drop-down menu. You can select either or both planes and place them in your mission. HOWEVER, what happens next depends on how your skins and models are configured.

If you have a single aircraft folder with multiple skins and each skin has its own **Model folder** and **model file** assigned to it, MB treats each one as a separate plane and you can fly both planes, each with its own skin, in the same mission.

On the other hand, if the two skins share the same model file then you can only use one of them at a time in any individual mission. Unfortunately MB doesn't tell you this when writing the mission. It allows you to select both and save the mission but when you go to fly it you will either get an error message and the mission won't load or it will load but only one texture will be used on both planes.

If you have a favorite plane with multiple skins and you choose to make copies of the plane's folder and place a different skin into each folder, make sure to give the copied plane both a new folder name as well as a new title name in the **aircraft.cfg** to insure that they all show up properly in both MB and FF.

Here's a guide to helping interpret the **mis** file names:

MIS File Name	Meaning
name=F4F4_Wildcat	Single model/single texture (aircraft folder name)
name=DB_F4U1C_Corsair	Multiple models/multiple textures - FIRST PLANE CHOSEN (folder name)
name=DB_F4U1C_Corsair.1	Multiple models/multiple textures - SECOND PLANE CHOSEN (folder name + #1)
name=SBD2_DAUNTLESS.1	Single model/multiple texture/same DP (folder name + #1)
name=P-38F Lightning Photo Recon	Single model/multiple texture/different DP (title name from aircraft.cfg)

If a plane with one model and multiple textures has both textures chosen it will only display one name, the one shown in [flightsim.1] of the aircraft.cfg file.

```
[runway.0]
id=8030
heading=66
units_across=2
base_lat=S9* 25.87'
base_lon=E160* 2.85'
base_alt=+55
takeoff_start_pos=-450,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=550,0
landing_stop_pos=-250,0
```

Following the list of objects used is a section for the airbases used in the mission. If there are no airbases designated **Take off** or **Land at base** there will be no airbase section in the mission file. All of the airbases used for takeoffs and landings, whether for the player or for AI flights, will be listed here; however, aircraft carriers used for take offs or landings will not be shown as they are not true airbases.

Each airbase used in the mission is numbered sequentially starting with **[runway.0]**. These numbers probably won't match the runway numbers assigned to the same bases in your airbases.dat file.

<u>The ID number shown is the ID number of the airbase in the airbases.dat file of the person who wrote the mission</u>. This is an important distinction to grasp. If you wrote the mission, the ID number will match the one for that airbase in your airbases.dat file. If the mission was one you downloaded from the Internet, the number will most likely not match your airbases.dat file. This is an important point as this mismatch of ID numbers can cause a Crash To Desktop (CTD) under the right circumstances. Refer to Chapter Fifteen for more information on this conflict and situations in which you may want to modify this field.

The last section in the mission file is the Strings section. You'll recall that there were references for the Summary, Objective and Intelligence strings at the very beginning of the mission file. Those pointers refer to the information contained here. The Strings section also contains the names of the aircraft, ships, vehicles and airbases used in the mission.

[strings]
"title_string"=**Luck of the Draw**
"summary_string"=**You and a wingman are flying a triangular grid out of Henderson. Keep alert and destroy any enemy targets that you encounter. The mission will end when you return safely to Henderson.**
"objective_string"=**Your plane was recently repaired and you want to take it out for a shakedown. You have planned a routine flight over Iron Bottom Sound.**
"intelligence_string"=**Enemy activity has been at a low point lately. You should have an uneventful flight but watch for some Japanese patrol boat activity around Florida Island.**
"player_aircraft_name_string"=F4F-4 Wildcat (Land Based)
"airfield_string"=Henderson
"name_string.0"=**Argo**
"name_string.1"=**Argo-1**
"name_string.2"=**Aoshima**
"name_string.3"=**Aoshima-1**
"name_string.4"=**Aoshima-2**
"name_string.5"=**Honda**
"name_string.6"=**Honda-1**
"name_string.7"=**PB-27**
"name_string.8"=**PB-75**

You can easily change the mission name or any of the briefing information. In fact, if you want to add a lengthy briefing text, one that is too long for entry in Mission Builder, this is exactly how you would do it. Simply open the **mis** file and insert the additional information in the appropriate section.

The most common thing you will probably edit here will be the aircraft names. In CFS2 every plane in the flight takes the name of the flight leader followed by a number. By editing the names you can have different names for each of your wingmen or the enemy planes. Deleting a name is the same as entering **No name** when selecting the moving object in Mission Builder. Some examples:

Instead of:	Edit to read:
"name_string.0"=**Argo**	"name_string.0"=**Hawkeye**
"name_string.1"=**Argo-1**	"name_string.1"=**Pappy**
"name_string.2"=**Argo-2**	"name_string.2"=**Tex**
"name_string.3"=**Argo-3**	"name_string.3"=**Slim**
"name_string.4"=**Argo-4**	"name_string.4"=**Butch**
"name_string.4"=**Honda**	"name_string.4"=**Saburo**
"name_string.4"=**Honda-1**	"name_string.4"=**Junichi**
"name_string.4"=**Honda-2**	"name_string.4"=**Shoichi**

The mission file is easy to hand edit and many of the changes, such as modifying the briefing information, are quicker and easier to do by hand than by opening and editing in Mission Builder. Some, such as personalized wingmen names, can only be done by hand editing. Obviously every field could be edited by hand but some just lend themselves to the task more so than others. Be sure to make a copy of the original **mis** file first in case something goes wrong, then edit away and test your changes in the game.

The DYN File At first glance the **dyn** file seems much less straight forward than the **mis** file. Generally speaking this is a file best modified in Mission Builder. However, there are some things you might want to hand-edit and knowing the structure of the file can't hurt.

The **dyn** file is divided into three sections, TRIGGERS, EVENTS, and ACTIONS. These correspond with the triggers, events, and actions those events perform that you set up in Mission Builder. By recalling what you wrote into the mission you can find your way through the **dyn** file easier.

Each trigger has an **id** number. This number will be referenced later in the Events section. The numbers do not run sequentially. Looking at a trigger entry you can generally figure out what it says. You can easily edit the name of any trigger. You could also change the waypoint numbers (for example changing the first trigger to arriving at waypoint #3 instead of #2). Realize that if you do this other elements of the mission may not play out properly. Another easy change would be to change the random numbers. In this case you could change "**Random 50**" to "**Random**

```
#
# TRIGGERS
#

trigger id 1
{
        name "Player arrives at WP #2"
        unit 6000
        waipoint_reached eq 2
}

trigger id 32
{
        name "Aoshima Destroyed"
        unit 6004
        unit_health eq 0
}

trigger id 31
{
        name "Random 50"
        random eq 50
}
```

75" then change **random eq 50** to read **random eq 75**. Now there would be a 75% chance of the event occurring instead of a 50% chance. When testing missions with multiple random percents I will occasionally hand-edit them in this manner to make sure each one is working properly rather than using Mission Builder.

The Events section lists all of the events you created. Like the triggers, each is assigned an ID number. Probably the only thing you would ever hand-edit here would be the event name. However, if you look at the format of the event you can see how it interacts with the triggers and actions.

```
# EVENTS
#

event id 4
{
        name "Spawn Honda"
        input id 1 and id 31
        action { id 33  id 34 }
        desc ""
}

event id 8
{
        name "Enable Warp ""Honda"""
        input id 4 and id 5
        action { id 38  id 39 }
        desc ""
}

event id 12
{
        name "Spawn first patrol boat"
        input id 9 and id 43
        action { id 77  id 78 }
        desc ""
}

event id 37
{
        name "Spawn Aoshima"
        input id 1 and (not id 4)
        action { id 35  id 36 }
        desc ""
}
```

Look at the first event, "Spawn Honda". Note that the inputs are id 1 and id 31. If you look above in the triggers you'll see that these inputs are "Player arrives at WP #2" and "Random 50". The actions are id 33 and id 34. If you scroll down to the ACTIONS section you'll see that these equate to disable warp and spawn unit 6001. "Unit 6001" refers to the unit number in the **mis** file for the flight we call "Honda.

Next, look at the event "Spawn Aoshima". Note that the inputs are id 1 and **(not id 4)**. ID 4 in this case refers to the earlier event "Spawn Honda". As you'll recall from when we wrote the mission, there is a 50-50 chance that Honda will spawn. If it doesn't, then Aoshima spawns (the action id36 spawns unit 6004 (Aoshima). With a little thought you can see how the **dyn** file puts into computer code what we entered as plain text in Mission Builder.

The last section of the **dyn** file contains the actions that each trigger/event combination generates. There are only a couple of things here that you might hand-edit. In the sample shown on the next page note **action id 38.** You'll see that is shows

type delay
5

This indicates a 5 second delay before the next called for action occurs. You can easily edit this delay time.

```
# ACTIONS
#

action id 33
{
	type disable_warp
}
action id 36
{
	type spawn
	unit 6004
}

action id 38
{
	type delay
	5
}

action id 39
{
	type enable_warp
}

action id 80
{
	type debrief_text
	string "Those Zeros caught you off guard but you handled yourself well.  Good work!"
}

action id 85
{
	type debrief_text
	string "Nice job bagging that patrol boat."
}
```

The section you are most likely to edit in the **dyn** file is the debriefing text. It is easier to open this file and correct spelling and grammar errors or change the text than it is to open Mission Builder and do it there. And, if you are going to use a very large debriefing that would cause Mission Builder to CTD, hand-editing is the only way you can insert that information.

Summary

The **mis** and **dyn** files are simple text files that can be easily edited. Some elements lend themselves to editing more readily than others. As you learn more about mission building don't hesitate to do some hand-editing. You might want to make a copy of the mission files first just in case you end up with something that won't work and you can't remember how to fix it!

Be sure to read the sidebar on page 123 that covers naming and saving **mis** and **dyn** files. All the editing in the world won't do any good if you don't save the finished product as a file usable by CFS2.

Now that you have a basic understanding of the structure and design of the mission files, we'll move along to some specific edits you can make which will make your missions more enjoyable to fly.

Chapter fourteen – weather & wingmen

Preflight
In this chapter we'll do four different hand edits: adding custom weather, flying as a wingman, having each plane in a flight be a different model, and increasing flights beyond eight aircraft.

Adding Custom Weather
The default weather in Mission Builder is clear skies but you can select cloud cover ranging from Light to Thunderstorm from the drop-down menu on the Mission Setup screen. Clicking **Advanced** will give you still more weather options, allowing you to set multiple layers of clouds, turbulence, level of precipitation, temperature and visibility among other things. With a little experimentation you can create some pretty complex weather. The advanced options allow you to set everything except wind.

On the other hand, Free Flight allows you to set all of the same weather parameters as Mission Builder with the addition of wind. The wind option under Free Flight allows you to set surface winds, gusts, turbulence, wind direction and shear strength. It sure would have been nice if Microsoft had included the same weather options in Mission Builder that they did in Free Flight. No problem, we can simply "borrow" the weather from Free Flight and insert it into our missions.

```
[weather]
visibility.0=16093.44,0.00,3048.00
vis_layers_num=1
cloud.0=9,914.40,2133.60,0.00,4,0,0,0,0.00,0,0
cloud_layers_num=1
temperature.0=1219.20,26.67,16.67,-17.78
temp_layers_num=1
wind_layers_num=0
wind_surface=609.60,0.00,0.00,0.00,0.00,0,0
```

Before adding custom weather to a mission (weather with custom wind in other words), let's look at how the mission and Free Flight weather match up. First design a mission and include whatever weather conditions you want using the advanced weather option in Mission Builder. This will create a **[weather]** section in your mission file. Save your mission and open the **mis** file using Notepad and you'll see something like the illustration to the left:

In this example I set up a 4/8 cloud cover of cumulous clouds between 3000 and 7000 feet with a 10 mile visibility, occasional turbulence, a temperature at 4000 feet of 80° and a 30° variation between night and day. I won't try to dissect all of the coding in the weather section, there are weather tutorials out there on the Internet that do that. We are focused solely on creating custom weather in Free Flight and importing it into our missions.

I then went into Free Flight and set up exactly the same weather conditions as I did in my mission. To find the Free Flight weather go to your main CFS2 folder and, using Notepad, open the file **cfs2.cfg**. Scroll down until you find the section **[Free Flight].** My Free Flight section is shown at right.

```
[Free Flight]
Clouds=2
Winds=0
TimeOfDay=1
visibility.0=16093.44,0.00,3048.00
vis_layers_num=1
cloud_layers_num=1
temp_layers_num=1
wind_layers_num=0
wind_surface=304.80,0.00,0.00,0.00,0.00,0,0
Time=14:30
AirfieldName=Henderson
AirfieldChecked=1
UserLon=E160* 2.56'
UserLat=S9* 26.00'
UserAlt=0
Aircraft=F4U-1A Corsair
Payload=0
AmmoLevel=100
FuelLevel=100
cloud.0=9,914.40,2133.60,0.00,4,0,1,0,0.00,0,0
temperature.0=1219.20,26.67,16.67,-17.78
```

Notice that there is additional information not related to the weather. Also notice that the weather information is not all in one place, the cloud and temperature information appears at the bottom of the section. Also note that the **wind_surface** information is different for the two files, even though both weather conditions were set up exactly the same. I have experienced this anomaly between the two files time and again and I have no explanation for it.

To add custom weather from Free Flight first create and save your mission using any cloudy or advanced weather, just so that you create a [weather] section. Next go into Free Flight and create the exact weather conditions you want. Then click **Fly Now**. You don't actually have to fly in Free Flight but you need to click *Fly Now* to save the weather changes in the cfs2.cfg file. When your flight starts, press **ESC** to exit Free Flight. Then open the **cfs2.cfg** file and scroll down to find your weather entry. Here's one I created that has high winds out of the west with lots of turbulence and wind shear. I have highlighted all of the relevant weather fields.

```
[Free Flight]
Clouds=3
Winds=3
TimeOfDay=1
visibility.0=4828.03,0.00,3048.00
vis_layers_num=1
cloud_layers_num=2
temp_layers_num=1
wind_layers_num=0
wind_surface=914.40,20.00,55.00,270.00,0.00,4,2
Time=14:30
AirfieldName=Henderson
AirfieldChecked=0
UserLon=E160* 2.56'
UserLat=S9* 26.00'
UserAlt=2222
Aircraft=A6M2 Zero
Payload=0
AmmoLevel=100
FuelLevel=100
cloud.0=9,762.00,2133.60,0.00,8,0,4,0,0.00,0,3
cloud.1=9,3048.00,4267.20,0.00,5,0,2,0,0.00,0,0
temperature.0=1219.20,26.67,16.67,-17.78
```

To add this weather to your mission simply copy the highlighted information and replace the lines of code in your **mis** file with the ones from the [Free Flight] section of the cfs2.cfg file. Save your changes and go flying. Your mission will now contain the custom weather you designed using Free Flight. It's very simple.

As a shortcut you can copy everything in the Free Flight section and paste that into your Weather section but I would not recommend it. Sometimes that works fine and other times it will cause the mission to fail to load.

There is one caveat to remember when adding custom weather; <u>if you edit the mission in Mission Builder you will lose all of the custom weather you added.</u> Even if you change something other than the weather, Mission Builder will remove your custom additions. Anytime I build custom weather into a mission I save a copy of just the weather information as a text file. Then, if I have to edit the mission later I can easily add the custom weather back in without having to reconstruct it again in Free Flight.

Don't hesitate to experiment with all of the other numeric codes in the weather entry. You may discover weather modifications that no one else has come across yet. Obviously there is a lot of data there that we don't appear to be using.

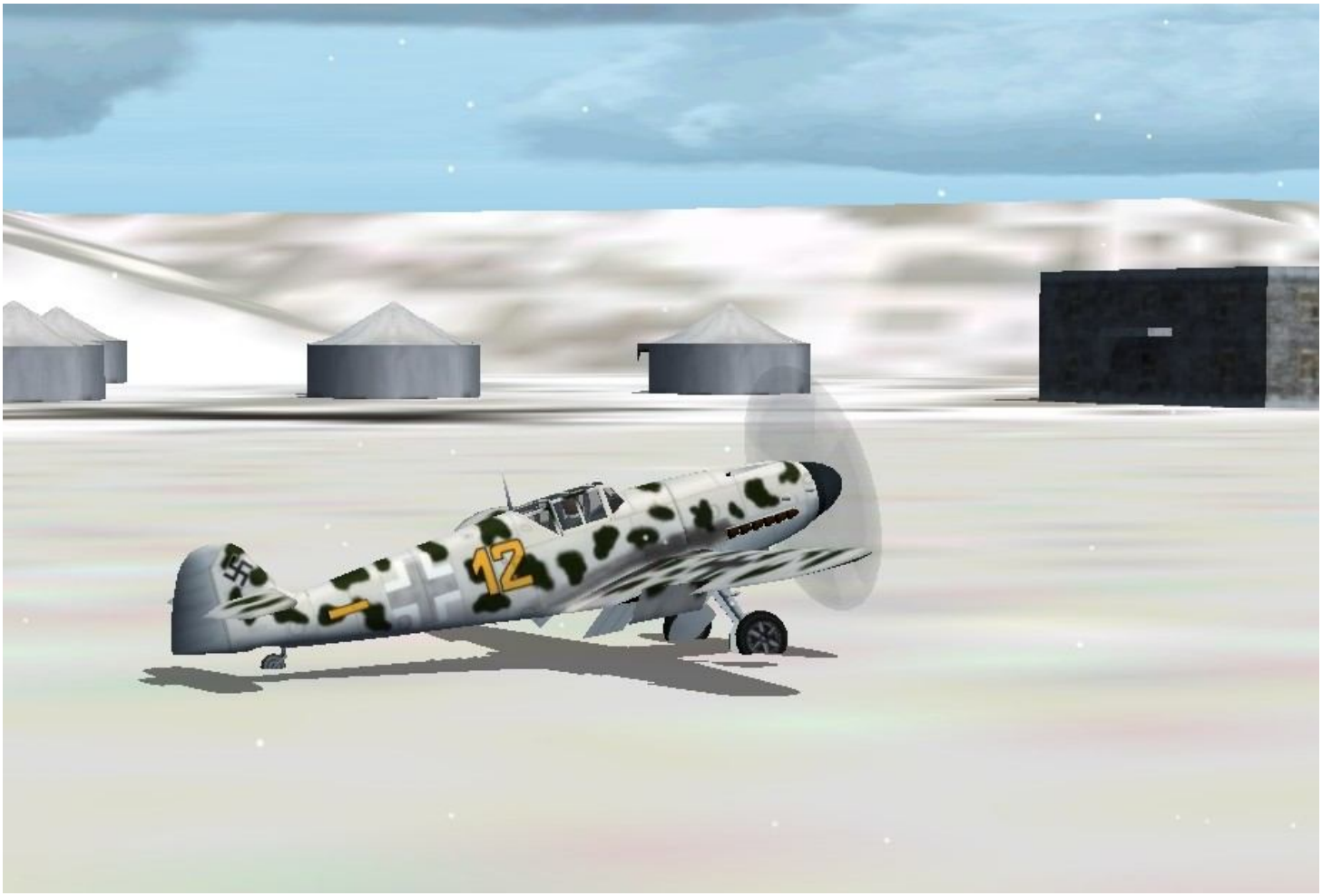
Snow That's right, snow! You can add snow to your missions. Although there seems little use for snow in the South Pacific, CFS2 does have the feature built into the system; probably a leftover from some prior flight simulator that CFS2 was based on. The snow is of poor quality, but if you have a need for it you can easily add it to your mission.

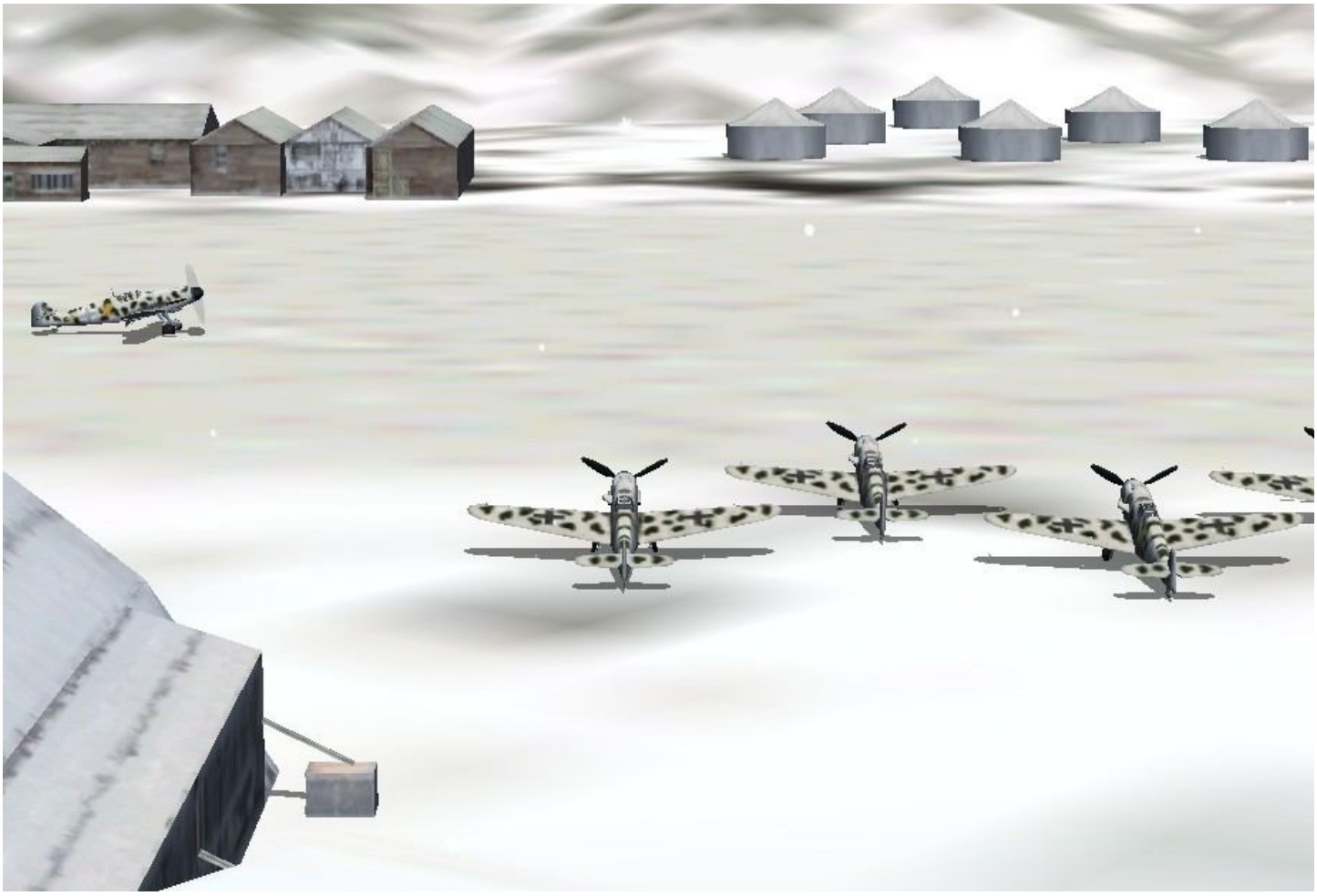
In the weather section of your **mis** file look at the first cloud line:

cloud.0=9,1499.62,2565.81,99.67,7,0,0,**1**,0.00,3,2

Note the number that I have highlighted in red. This character represents the type of precipitation in the mission, 0 is none, 1 is rain, and **2 is snow**. Change this number to a "2" and you'll have snow.

The "snow" appears as white blobs, similar to rain only less distinct. If you look closely in the two illustrations below you can see the effect. No matter how heavy you set the precipitation you'll only get a small amount of snow falling, less than if you had created rain. The snow won't stick to your plane and it won't accumulate on the ground. Still, if you have winter textures installed such as in these illustrations, it is better than nothing.





Like custom weather imported from Free Flight, if you edit the mission in Mission Builder you will lose your snow and have to add it back in.

One final word about weather, the weather you set up in your mission will apply world wide. There is no way to create two different weather conditions in the mission, say a thunderstorm over your base and clear skies over the target. What you get at one location will be the same weather everywhere. See the following sidebar, *Real Time Weather* for an interesting alternative.

Real Time Weather

While the weather you build into your missions will be the same worldwide, there is a way to get constantly changing weather as you fly around the CFS2 world. **ActiveSky** is a wonderful flightsim add-on that gathers actual weather information from around the world and simulates that weather in real time as you fly over that spot. In other words, if it's raining over New Guinea right now, it will be raining over New Guinea as you fly there in CFS2. Unless weather is an integral part of my mission, I always use **ActiveSky**. You can't control the weather in **ActiveSky**, but you can't control it in real life either. As a result each time you fly you may experience changing conditions; just like in real life! I have a very old version, 1.4. I've heard, but haven't confirmed, that newer versions of **ActiveSky** won't work on old flightsims like CFS2. If that's the case search Google and see if you can find an older version. You'll be glad you did. ☺

Flying as a Wingman

Have you ever wanted to fly as a wingman? Staying in formation can be tricky, especially when flying a heavy bomber. Not only does it add a new and challenging dimension to your missions, but it'll help you develop your piloting skills as well. Switching places with your wingmen is one of the quickest and easiest hand-edits you can make in CFS2. Here's how.

```
[formation.0]
id=6000
type=air
waypoint_path_id=5000
directive=8
players_formation=1
formation_type=1
points=2
unit_id_position.0=9078,0,0.0,0.0,0.0
unit_id_position.1=9079,0,-300.0,-600.0,0.0
unit_id_position.2=9080,0,300.0,-600.0,150.0
unit_id_position.3=9081,0,600.0,-1200.0,150.0
unit_id_position.4=9082,0,1200.0,-1200.0,300.0
unit_id_position.5=9083,0,900.0,-1800.0,300.0
unit_id_position.6=9084,0,1500.0,-1800.0,450.0
unit_id_position.7=9085,0,1800.0,-2400.0,450.0
```

Open up the **mis** file and scroll down to the player's formation, it may or may not be **[formation.0]** but it will say **players_formation** on the sixth line (see the illustration). You'll recall from Chapter Thirteen, the player's aircraft is listed first, 9078 in this case. To swap positions with any other plane in the flight simply copy the player's number to the position you want to fly in and copy that aircraft number to the first position.

For example, if we wanted to fly in the sixth position, simply replace **9083** with the player's number **9078** and vice versa. Save the mission and you will now be flying in the sixth position instead of leading the flight.

```
[formation.0]
id=6000
type=air
waypoint_path_id=5000
directive=8
players_formation=1
formation_type=1
points=2
unit_id_position.0=9083,0,0.0,0.0,0.0
unit_id_position.1=9079,0,-300.0,-600.0,0.0
unit_id_position.2=9080,0,300.0,-600.0,150.0
unit_id_position.3=9081,0,600.0,-1200.0,150.0
unit_id_position.4=9082,0,1200.0,-1200.0,300.0
unit_id_position.5=9078,0,900.0,-1800.0,300.0
unit_id_position.6=9084,0,1500.0,-1800.0,450.0
unit_id_position.7=9085,0,1800.0,-2400.0,450.0
```

There are four things to keep in mind when flying as a wingman. First, your callsign won't change unless you also edit the callsigns at the end of the **mis** file. In this example we would still show as *Agate* instead of *Agate-5*. Second, even though you are flying as a wingman you are still in control of the radio commands. The new flight leader will not give the commands "Split", "Attack", etc. Although only you can give these commands, the wingmen won't obey them. So, if you are flying in the sixth position and give the command to "Split" nothing will happen. Third, if you make any modifications to the mission in Mission Builder the game will reset you to the lead position and you will have to edit the mission again to return to your wingman's slot.

Finally, if your mission starts with your flight taking off, don't waste any time getting airborne. When you are the flight leader no one moves until you do, so you can take your time getting started. When you are flying as a wingman the flight leader's engine is running and he is ready to go. As soon as he starts to move, the remainder of the flight begins to roll. If you have an aircraft behind you and you don't get started on time they will run into you and both of your planes will explode, killing you and ending the mission. When flying as a wingman get your engine started right away and roll off the line as soon as it's your turn.

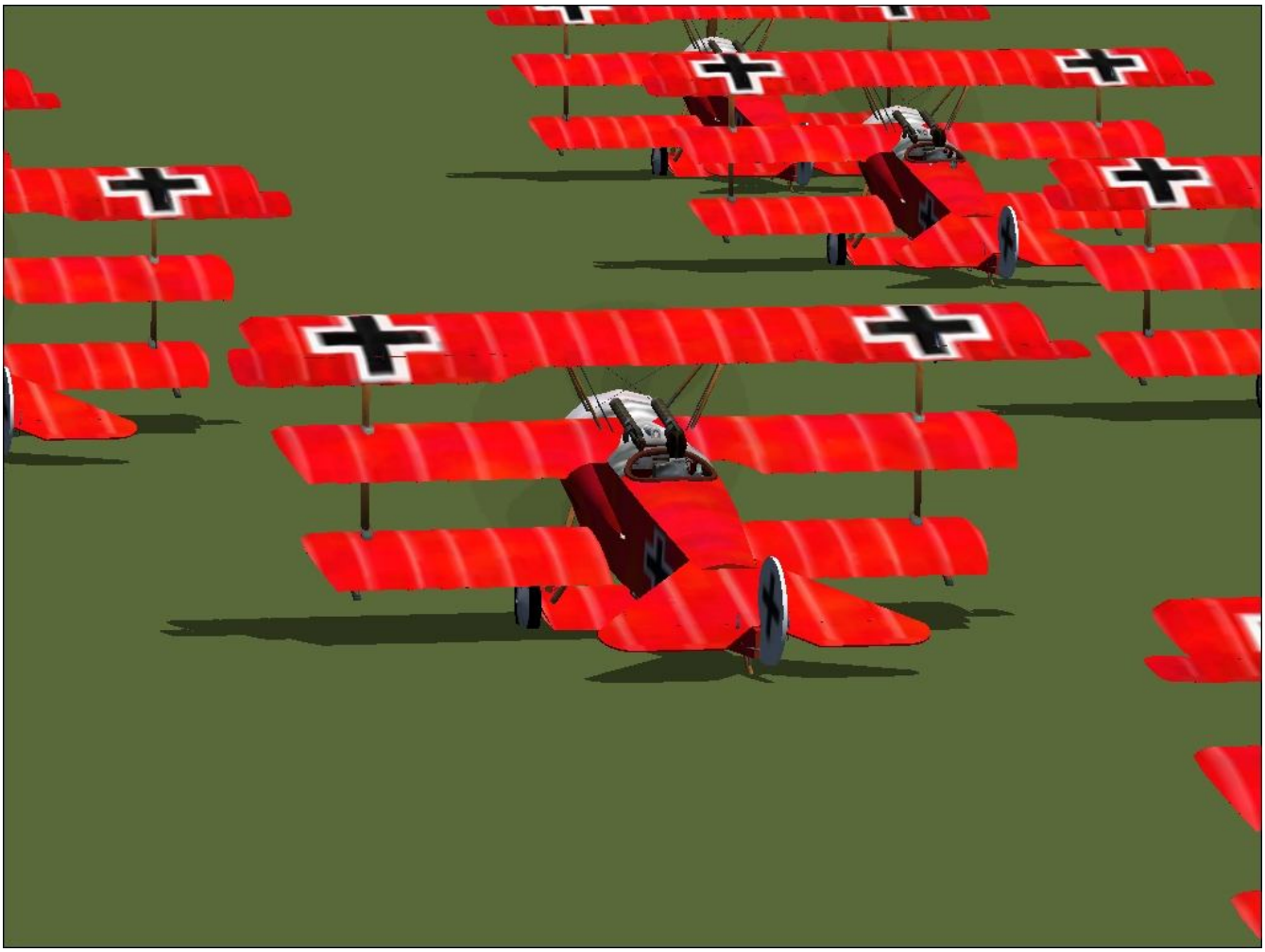
☛ Another quick and easy edit is to change the starting altitude of your plane. Mission Builder won't allow you to set a starting altitude below 164 feet (the arrows will only go down to 500' but you can type in numbers as low as 164). That's a little high if you're trying to write a mission where you come in at wave-top level to avoid radar. To change the starting altitude simply open the **mis** file and scroll down to waypoint section for the player's flight. On the **waypoint.0** line the first number after the longitude and latitude is the starting altitude. Simply change this number to something more appropriate, save your mission and go flying. In the example below I've changed my starting altitude to 50 feet, much better for what I have in mind.

waypoint.0=1,0,S0* 57.22',E144* 28.47',**+50**,,,,175,,0,0,1

Two cautions, if you edit the mission in Mission Builder it will reset your altitude to 164'. And be careful setting it too low over ground. If you have custom mesh scenery installed you might find yourself starting the mission inside of a mountain. Ouch!

Having your Wingmen all fly Different Planes – A True Flying Circus

When you create a mission using wingmen in CFS2 all of your wingmen fly exactly the same plane that you do. While that isn't a big deal, when you have additional aircraft with other textures it would be nice if you could have a flight where each of your wingmen had his own markings. And, if you have a World War One install of CFS2, which would you really rather have for a flying circus, this:



or this:

With a little bit of editing you can have each of your wingmen fly a different aircraft. It's worth the effort and it will really dress up your missions.

Begin by writing any mission with multiple wingmen. For this exercise I will use our first mission, *Beginner's Luck*. I've added four wingmen, modified the briefing slightly to reflect the additional planes and saved the mission as *Flying with Friends*. Once you are finished, save the mission and open it in Notepad.

Scroll down to the player's formation section. You'll recall from the previous chapter that each unit in the player's formation is listed starting with the player's aircraft, usually, but not always, [unit.0] . Each aircraft has a unique **id** number as well. In my mission they start with **id=9080**, your numbers may be different. Notice that the next line is the same for each aircraft **type_id=3814**. Again, your number here may be different. The key thing is that this number is the same for each aircraft in our flight. See the illustration on the next page.

```
[unit.0]
id=9080
type_id=3814
name=%name_string.0%
is_player_aircraft=1
allegiance=3
skill=1
payload=0

[unit.1]
id=9081
type_id=3814
name=%name_string.1%
allegiance=3
skill=1
payload=0

[unit.2]
id=9082
type_id=3814
name=%name_string.2%
allegiance=3
skill=1
payload=0

[unit.3]
id=9083
type_id=3814
name=%name_string.3%
allegiance=3
skill=1
payload=0

[unit.4]
id=9084
type_id=3814
name=%name_string.4%
allegiance=3
skill=1
payload=0
```

☛ This technique works with AI flights as well. Just follow the same steps using the AI units instead of the player's flight.

Also, you don't have to change the entire flight. You can choose just one or two aircraft instead of all of them.

If you scroll down to the bottom of the **mis** file you'll see the section where all of the objects used in the mission are listed. Find the section listing the Wildcat.

```
[unit_type.1]
type_id=3814
unit_family=1
name=F4F4_WILDCAT
```

Note that here too it shows **type_id=3814**. In plain English, what this entry is telling us is that every unit with a *type_id* of 3814 will be an F4F-4 Wildcat. That's our key. All we need to do to change every plane in our flight to a different model is to assign each plane a different type_id then create a corresponding entry at the bottom of the list of objects.

The first step is to assign each of our wingmen a new type_id. Simply increase the first wingmen's number to **3815**, the second to **3816**, etc. until all have been changed. When you're finished, the player's type_id will remain 3814 and the wingmen's will be numbered 3815 through 3818. See a partial list below:

```
[unit.0]                PLAYER
id=9080
type_id=3814
name=%name_string.0%
is_player_aircraft=1
allegiance=3
skill=1
payload=0

[unit.1]                1st WINGMAN
id=9081
type_id=3815
name=%name_string.1%
allegiance=3
skill=1
payload=0

[unit.2]                2nd WINGMAN
id=9082
type_id=3816
name=%name_string.2%
allegiance=3
skill=1
payload=0
```

Now scroll down to the bottom of the **mis** file to where the list of objects used appears. You'll notice that we have an entry for the Zeros, an entry for our Wildcat, and an entry for the torpedo boat. If we had lots of other planes, ships, or infrastructure objects they would all be listed here. We need to add new entries to the bottom of the list for each of our wingmen that we are changing. We will add four new entries, one for each wingman. The entries need to follow the correct format. The easiest way to create these entries is to copy the entry for the player's Wildcat and paste four copies below the last entry on the list.

```
[unit_type.0]
type_id=3802
unit_family=1
name=A6M5_ZERO

[unit_type.1]
type_id=3814
unit_family=1
name=F4F4_WILDCAT

[unit_type.2]
type_id=6052
unit_family=5
name=shj_torpedo_boat

[unit_type.1]
type_id=3814
unit_family=1
name=F4F4_WILDCAT

[unit_type.1]
type_id=3814
unit_family=1
name=F4F4_WILDCAT

[unit_type.1]
type_id=3814
unit_family=1
name=F4F4_WILDCAT

[unit_type.1]
type_id=3814
unit_family=1
name=F4F4_WILDCAT
```

On the left you see the way your **mis** file will look after copying four new entries for the wingman. On the right you see the finished version showing the information that needs to be edited.

First: Continue the numbering sequence for the [**unit_type.*n***] numbering format. Note that our wingmen will be units 3-6.

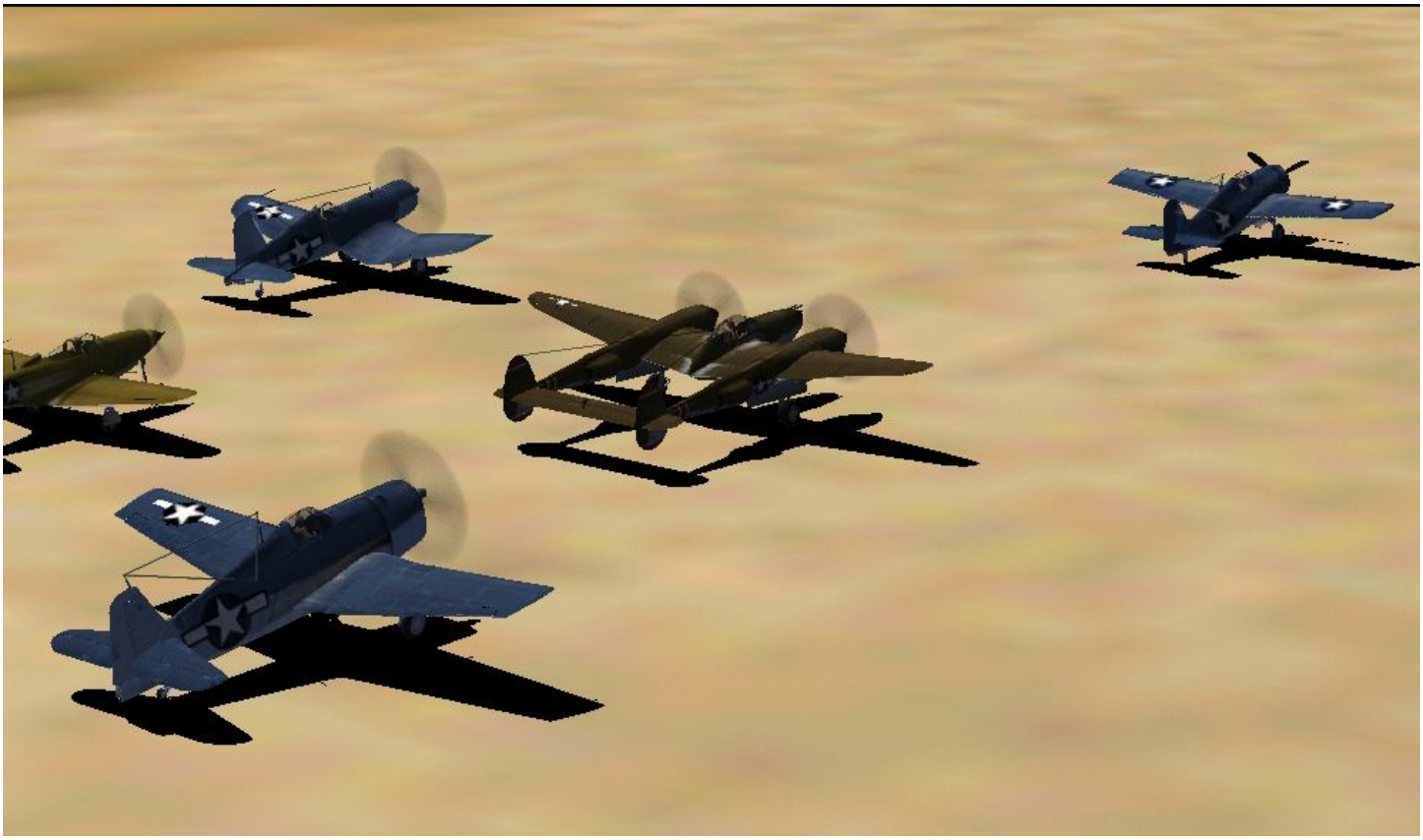
Second: Type in new **type_id** numbers to match the ones you entered above. These link the new aircraft types to the planes in your flight.

Third: Enter the folder name for each different type of aircraft you want in your flight. (Generally you will use the folder name. Refer to the sidebar *What's in a Name* in Chapter Thirteen for other alternatives).

That's all there is to it. Save your mission and go flying. When the mission starts, go to Spot View and you'll see something like the illustrations on the next page. This technique takes a little time but it can sure dress up your flights. Try it; I think you'll be impressed.

```
[unit_type.0]
type_id=3802
unit_family=1
name=A6M5_ZERO

[unit_type.1]
type_id=3814
unit_family=1
name=F4F4_WILDCAT

[unit_type.2]
type_id=6052
unit_family=5
name=shj_torpedo_boat

[unit_type.3]
type_id=3815
unit_family=1
name= F4U1A_CORSAIR

[unit_type.4]
type_id=3816
unit_family=1
name= P38F_LIGHTNING

[unit_type.5]
type_id=3817
unit_family=1
name=P39D_AIRACOBRA

[unit_type.6]
type_id=3818
unit_family=1
name=F6F3_HELLCAT
```

As with many of the other hand-edits, if you edit any portion of this mission using Mission Builder you will lose all the changes you made and everyone in your flight will be back in a Wildcat. I'd recommend that you save a copy of the changes you made so you can paste them back in easily should you need to modify the mission at a later time.



Flights of More Than Eight Planes

The Mission Builder limits the size of any flight (yours or AI) to eight planes. While eight is more than enough for most occasions, there may be times when you want to expand that size to 10, 12, 20, or more. With some hand editing you can make your flight any size you want. Just remember, they are still your wingmen and they will all follow your waypoints and obey your commands.

Begin by locating the units in the player's formation. Usually this is the first formation listed, but in complex missions it could be far down on the list. Look for the words "**players_formation=**". In a really large mission, you might find it easier to use the **Find** function in Notepad to locate it quickly. The formation information is located below the unit information for each flight, so once you find the player's formation you will want to scroll up to find the individual units. In our example we have a simple mission with only one flight of eight planes so the player's formation starts with [unit.0] and runs through [unit.7]. Using the format of the first seven as an example, add as many additional units to the formation as you would like. In this example I show you the last of the original player's wingmen [unit.7] and I've added three more, units 8, 9, and 10. To simplify things, everyone in our flight will be flying the same type of aircraft so all of the **type_id**'s will be the same.

```
[unit.7]
id=9162
type_id=3954
name=%name_string.7%
allegiance=3
skill=3
aggression=2
payload=0

[unit.8]
id=9163
type_id=3954
name=%name_string.8%
allegiance=3
skill=3
aggression=2
payload=0

[unit.9]
id=9164
type_id=3954
name=%name_string.9%
allegiance=3
skill=3
aggression=2
payload=0

[unit.10]
id=9165
type_id=3954
name=%name_string.10%
allegiance=3
skill=3
aggression=2
payload=0
```

Note that the unit number, ID number, and name_string number must increase by one each time. The **name_string** number should always match the unit number.

Next scroll down to the formation section. You will need to add an additional line for each plane that you added to the flight.

```
unit_id_position.0=9155,0,0.0,0.0,0.0
unit_id_position.1=9156,0,-300.0,-600.0,0.0
unit_id_position.2=9157,0,300.0,-600.0,150.0
unit_id_position.3=9158,0,600.0,-1200.0,150.0
unit_id_position.4=9159,0,1200.0,-1200.0,300.0
unit_id_position.5=9160,0,900.0,-1800.0,300.0
unit_id_position.6=9161,0,1500.0,-1800.0,450.0
unit_id_position.7=9162,0,1800.0,-2400.0,450.0
unit_id_position.8=9163,0,-100.0,-100.0,0.0
unit_id_position.9=9164,0,-200.0,-500.0,0.0
unit_id_position.10=9165,0,200.0,-500.0,150.0
```

In the example above you can see where I've added our three planes. Make sure that the unit and id numbers match those that you entered above. Also, make sure that the relative position information does not match that of any shown in the original flight or you'll have two planes trying to start in the same spot causing those planes to crash at mission startup. Refer to Chapter Thirteen, Page 124 for more on adjusting a plane's relative position within a flight.

Finally, scroll down to the bottom of the mission file and add new **name_string** lines for your additional wingmen.

```
"name_string.0"=Argo
"name_string.1"=Argo-1
"name_string.2"=Argo-2
"name_string.3"=Argo-3
"name_string.4"=Argo-4
"name_string.5"=Argo-5
"name_string.6"=Argo-6
"name_string.7"=Argo-7
"name_string.8"=Argo-8
"name_string.9"=Argo-9
"name_string.10"=Argo-10
```

Save the mission and the next time you fly you'll have a flight of ten instead of eight.

In theory you could use this method to create a flight of one hundred B-17's all controlled by you. They would follow your commands and bomb the target you bracketed. They would also turn with you and try to take off and land with you. I say "in theory" because I haven't tried doing a flight that large, but technically I see no reason why it wouldn't work, unless you ran into a problem with the mission file itself being too large to load.

Mission Containing Player's Flight and AI Flight - The problem is a little more complex when you have AI planes in your mission. Even if you only want to increase the size of the player's flight you must still modify the AI flight information to get this technique to work.

Using our example above we'll say that our mission also included a flight of Zeros. After you've made the changes described above to the player's flight, locate the list of the units in the AI flight.

Note that the original unit and id numbers of the AI flight match the newly added player wingmen. These old numbers must be changed. Renumber the AI flight starting with the next numbers in sequence from your modified player flight.

Original AI numbers

```
[unit.8]
id=9163
type_id=3806
name=%name_string.8%
allegiance=4
skill=2
aggression=2
payload=0

[unit.9]
id=9164
type_id=3806
name=%name_string.9%
allegiance=4
skill=2
aggression=2
payload=0
```

Renumbered AI flight

```
[unit.11]
id=9166
type_id=3806
name=%name_string.11%
allegiance=4
skill=2
aggression=2
payload=0

[unit.12]
id=9167
type_id=3806
name=%name_string.12%
allegiance=4
skill=2
aggression=2
payload=0
```

Since we stopped numbering our new player's wingmen at [unit.10] and id=9165, begin numbering your AI flight at **[unit.11]** and **id=9166**. It's simple really, all of the planes (units) must be numbered in sequence. Spend a few minutes studying the format and it will come to you. Make sure that you also change the number_string to match the new id numbers.

Next change the id numbers in the AI formation to match the new ones.

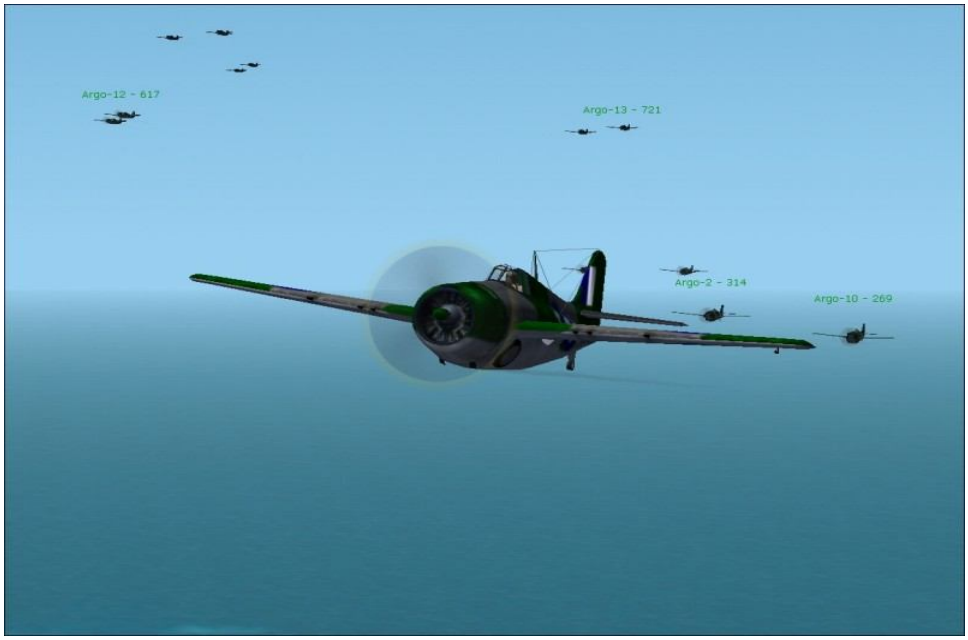
```
unit_id_position.0=9166,0,0.0,0.0,0.0
unit_id_position.1=9167,0,-300.0,-600.0,0.0
```

Finally, drop down to the bottom of the mission and add the correct name_string information for the AI flight.

```
"name_string.8"=Argo-8
"name_string.9"=Argo-9
"name_string.10"=Argo-10
"name_string.11"=Abaniko
"name_string.12"=Abaniko-1
```

The more additional units you have the more complex it becomes but if you stay focused you can make the modifications and increase your formation sizes. Make sure you follow all of the steps above to insure success.

As with many other hand edits, if you open the mission in Mission Builder and make any changes you will lose all of your modifications and the mission will revert back to just seven wingmen.



Included with the Handbook are two sample missions in PDF format showing this procedure. One is a simple mission with 8 Wildcats and 8 Zeros, the other is the same mission modified to have 16 Wildcats. In the modified version all of the new information that has been added or changed is in **bold** so that you can see it easily. If you wanted to, you could copy the text from the PDF document into Notepad and save it as a mission file for use in CFS2.

Summary
While we took each of these edits individually, you could combine any or all of them into a single mission. If you wished, you could fly as the wingman in a flight of 12 different kinds of planes with custom weather. Far fetched perhaps, not really. Consider designing a bombing mission over Germany. You could build a flight of 32 B-17's, each with it's own skin (32 different aircraft). You could fly as any one of the 32 bombers and add in some light cloud cover and winds. It would be like being on an actual WWII bombing run. You try to stay in formation and drop your payload when the group leader drops his. The trick is to follow the instructions carefully and make all of the appropriate changes.

I mentioned it many times in the chapter but it bears repeating one last time. If you make any of these hand edits and later open the mission in Mission Builder you will lose your changes. That could represent over an hour of editing down the drain with just a couple of clicks. I would recommend that you create a folder just for your hand edited missions. Place one copy of the mission in this folder; the original goes into the MISSIONS folder in CFS2 so you can fly it. Then if you should ever inadvertently lose your changes, you can restore the mission from the spare copy.

Taking Off and Landing in the Same Direction

I'm sure you've noticed that in CFS2 you take off in one direction and land in the opposite direction. So if you leave to the north, you land to the south. In the real world, we always try to take off and land into the wind, which means that most of the time we are taking off and landing in the same direction. This isn't an issue in CFS2 since it doesn't include wind in the missions. But, if you've added custom wind such as described above, and you want to be more realistic in your flying, here's an easy hand edit to fix this.

Open your mission file and scroll down to the runway for your flight. Look near the bottom for the line:
landing_start_pos=5000,350

Change the 5000 to **-10000** so it reads: landing_start_pos=**-10000**,350. Save your mission and you're done. By changing the landing start position from plus 5,000 to minus 10,000 you have reversed the direction you'll be facing when you warp back to your base. You'll be lined up on approach a couple of miles from the runway facing the same direction as when you took off.

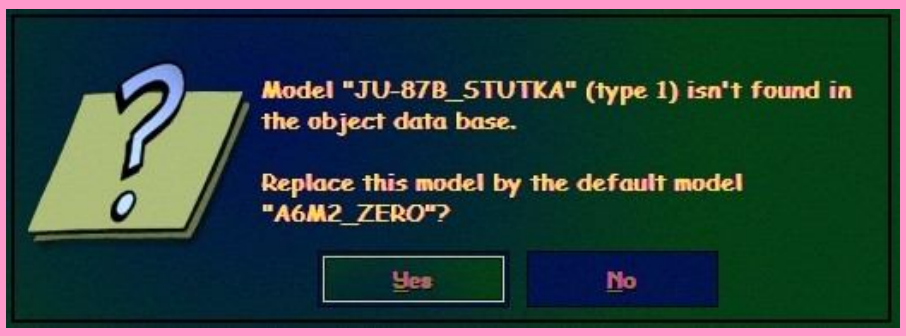
This edit will effect only this one mission. If you want to make a permanent change for this airbase, open the **airbases.dat** file in the **INFO** folder. Find the entry for the base and make the change there. Now you will always take off and land in the same direction from that base (unless you edit the mission again as we just described).

Be sure to enter **-10000**. If you simply enter a minus sign in front of the 5000 you will end up facing the right direction, but you'll be directly over the runway and unable to land without turning around and making another approach. ☺

Converting CFS1 Missions to CFS2

Can you modify CFS1 missions for use in CFS2? Yes. Should you? Well that's a different matter altogether.

Anytime you try to load a mission in Mission Builder that uses planes, ships, or infrastructure that you don't have in your game you will see an error message that looks like this:



The error message lists the object that it can't find and asks if you want to replace it with the stock Zero. Now you're in a box. If you press **OK** it moves on until it finds the next missing item or the mission loads. If you press **No** you get this error message and the mission won't load:

The problem with CFS1 missions is that you are probably missing most, if not all, of the items called for in the mission. To load them you have to replace everything with the Zero. Now you end up with a mission full of Zeros and you don't know what the original object was. It might have been a Spitfire, or a Messerschmitt, or battleship, or a tent. You have no idea since every replaced object is now a Zero. You're left to guessing what was what.

Of course, you could open the mission first using Notepad, scroll down to the list of objects used, and replace the ones you don't have with similar ones from your installation. That will at least insure that the mission loads and you can edit from there. Still, you might find the results pretty awful.

The bottom line is this; if you've come this far in the Handbook then you're a skilled and versatile mission builder. In the long run you'll find it much quicker (and easier) to write new European Theater of Operations (ETO) missions from scratch rather than trying to load and convert old CFS1 missions. There are a wealth of great add-ons out there to create a vivid ETO within which to set your missions and campaigns. Spend the time necessary to download and create a dedicated ETO installation then write your own challenging and exciting missions using the techniques you've learned here. In the long run you'll be more satisfied and have better quality missions to fly as well! ☺

Chapter fifteen – how to take off from any spot on the planet

Preflight
In CFS2 you can land anywhere in the world. However, using the Mission Builder, you can only take off from designated spots. Those spots are runways or seaplane bases. They're fine 90% of the time, but sometimes you might want to design a mission where your plane takes off from a different location. For example, maybe you'd like to start a mission from inside an aircraft revetment or hanger and have to taxi to the runway before taking off; or maybe you want to fly a seaplane from some uninhabited, Pacific atoll; or perhaps you want to take off from a French hay meadow in the dark of night flying your Lysander back to England just ahead of approaching Me109's.

Whatever the scenario, there are times when it would be nice to be able to take off (or have AI planes take off) from some spot other than the designated airbases. In this chapter we will cover how to create a take off point anywhere on the planet. This method is not intended for creating a new, permanent airbase. If you want to use the same spot over and over again then you should make an entry in the airbases.dat file (see Chapter Eighteen for more information on creating permanent entries in this file). The technique we will cover here is for those situations where you want to create unique take off points on a mission-by-mission basis.

Creating a Take Off Point Without the use of the *airbases.dat* File
The Mission Builder map indicates valid take off locations with an '**X**'. Blue 'X''s are for friendly bases, red for enemy bases and grey, the default color for all CFS2 bases, denotes a neutral base. The location of the 'X' on the map is determined by an entry in the **airbases.dat** file found in the INFO folder within your main CFS2 folder. The airbases.dat entry tells the game that a specific location is a valid place for using the **Take Off** or **Land at Base** commands. When the airbases.dat entry coincides with the location of installed scenery (as it does for all of the stock airbases) you can begin the game by taking off from a runway, complete with hangers, revetments, and other associated infrastructure.

Any airbase scenery that is active in the game but doesn't have a corresponding airbases.dat entry won't show as an 'X' on the map. Conversely, you could have an airbases.dat entry for a location where there is no scenery and that spot would show as an 'X' in Mission Builder. That last example is how we make seaplane bases. Pick a spot just offshore and create an airbases.dat entry for it. Now it will show as a takeoff point in Mission Builder and you can fly your PBY from it. There is no scenery there, only the ocean, but it is a valid take off point none the less.

Think of the Mission Builder as a big map containing scenery, and the airbases.dat file as a plastic overlay containing the 'X''s that you lay on top of the map. As you add or remove entries from the airbases.dat file (the overlay) you are adding and removing where the 'X''s appear on the map. The two are independent entities but they work together to create the sensation of taking off from an airbase. You can change either the map or the overlay without effecting the other. If you delete the scenery for an airbase but fail to delete the associated airbases.dat entry you can still write missions that take off from that spot. The only problem is that when you get there you'll be all alone, no runway, no buildings, no nothing.

OK, so if all you have to do to create a Take Off spot in Mission Builder is to add an entry to the airbases.dat file, why this chapter? Well, you may want to write a one of a kind mission, such as where an enemy seaplane takes off from a hidden cove as you approach. You don't plan to ever use

that spot again so why waste space in the airbases.dat file with a take off location that you only use once? Additionally, if you have an older system, your airbases.dat file will be limited to 65kb, or about 242 entries.

The Mechanics of a Take Off Whenever you write a mission that uses either the **Take Off** or **Land at Base** commands, the Mission Builder reads the data for the runway used from the airbases.dat entry and writes it directly into the mission file. It copies all of the data except for the runway name, number and allegiance. Note that the Mission Builder assigns its own sequential number to the runways used in the mission starting with [runway.0].

airbases.dat entry for Henderson

```
[runway.76]
id=8030
name=Henderson
base_lat=S9 25.87
base_lon=E160 2.85
base_alt=55
heading=66
allegiance=0
units_across=2
takeoff_start_pos=-450,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=550,0
landing_stop_pos=-250,0
```

Mission Builder entry for Henderson

```
[runway.0]
id=8030
heading=66
units_across=2
base_lat=S9 25.87
base_lon=E160 2.85
base_alt=55
takeoff_start_pos=-450,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=550,0
landing_stop_pos=-250,0
```

It's to our benefit that Mission Builder works like this. If you download one of my missions you need to know where to take off from. The mission file defines that location exactly, using longitude and latitude. Why not use the runway ID number of **8030** instead and simply have CFS2 look up the location in the airbases.dat file? Because your Henderson and my Henderson may not have the same ID number. That probably won't be the case with the stock runways but it is very frequently the case with add-on airbases. Also, if you didn't have a runway **8030** entry at all, your CFS2 would never find the starting location and the mission wouldn't work. By writing the take off and landing coordinates directly into the mission file, CFS2 always knows exactly where you should be.

This is why you can download an Aleutian Islands mission and fly it even if you don't have any Aleutian Islands airbases installed. The mission will still work. You won't see the runway or hangers or anything else that is included in the scenery package but you will see the other planes, ships, vehicles and infrastructure added by the designer of the mission.

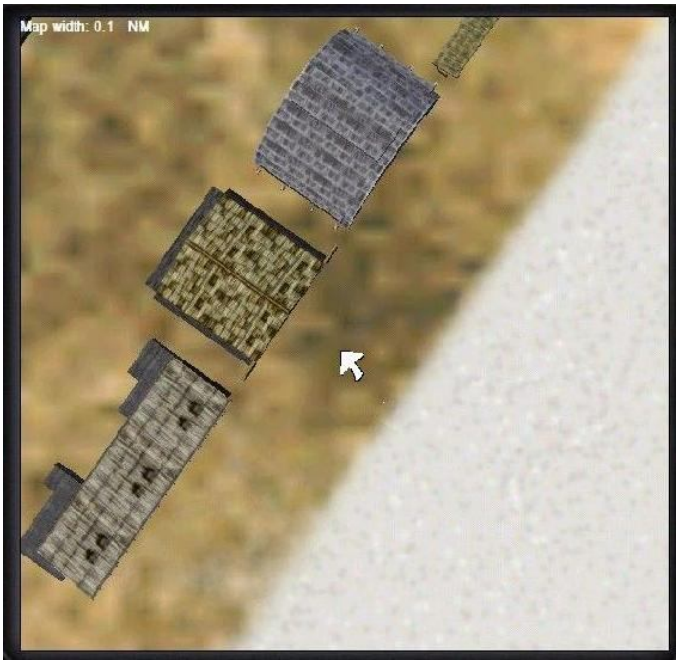
Since the exact take off location is incorporated into the mission file we can tweak that location to suit our needs for any individual mission. Simply change that location and you change your take off point.

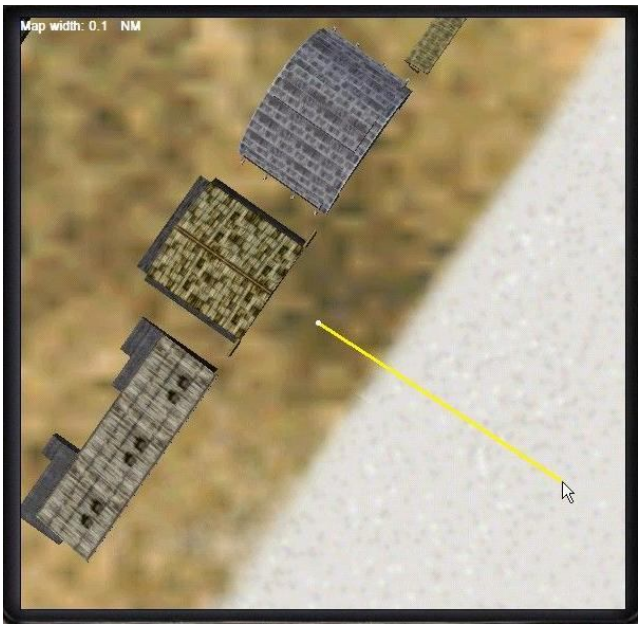
Changing your Take Off Location Let's assume that you want to fly a mission out of Wake Island and instead of starting at the end of the runway you'd like to start near the hangers on the west side. Begin by selecting Wake as the location where you want to start your mission. After you've chosen a plane, go to the Route Tab and choose **Take Off** for your plane's action. Your screen should look like the illustration on the next page.

Using the Zoom button, begin zooming in on Wake until you see an image like the one shown at the right. It will take about 5 clicks to get to that level. Notice the collection of hangers on the left side of the main runway about in the middle of the screen. Center your Zoom Tool on them and continue to zoom in until you get to a map width of 0.1 mile and the hangers fill the screen. See the illustration on the next page for an example.

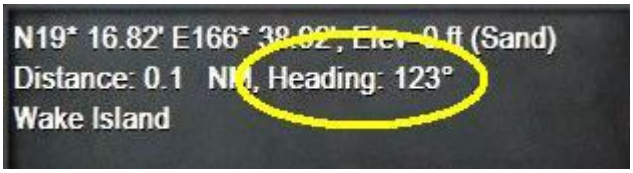
Now that you are zoomed in close to the hangers, place your cursor directly in front of the center hanger (the one with a thatch roof) about mid-way between the hanger and the runway and read the longitude and latitude shown in the lower left. The spot you choose will be your new take off point.

Notice that in this example the longitude and latitude reads **N19* 16.85' E166* 38.87'**. Write this number down! You will need it later. Also make a note of the elevation at this spot. Although in this example it won't be necessary due to our close proximity to the original take off point, it is important if the new take off point is at a significantly different elevation than the old one.





Without changing your zoom setting you will now take the heading measurement for your new take off point. Left click and drag your cursor toward the runway. A yellow line will follow the cursor's path and you can read the heading just below where you read the longitude and latitude. In this example our heading is 123°. Write this down as well.



Now you can zoom back out and finish designing your mission. Add any additional planes, ships, vehicles, infrastructure, etc. For this exercise I suggest that you use only the Player Aircraft and just lay out waypoints where you take off and fly around the island. Complete the process of designing and testing your mission. Note that when you test the mission you will be taking off from the runway, not starting in front of the hanger. That is OK for now as you are testing the mechanics of the mission to see that it works properly. Once you change the start location you won't be able to edit the mission in Mission Builder again without having to manually re-edit the start location. To save time it is easier

to do all of your testing starting from the runway. When the mission is complete, name it and save it as you would any other mission.

Open the **mis** file using Notepad and scroll down until you see the runway information for Wake. An example is shown below with the fields highlighted that we will change:

[runway.0] ……..	Sequential runway number for this mission
id=8015 ……..	Matches ID number shown in my *airbases.dat*
heading=34 ……..	Change this to **123**
units_across=2	
base_lat=N19* 17.04' …….	Change this to **N19* 16.85'**
base_lon=E166* 39.07' …….	Change this to **E166* 38.87'**
base_alt=+0 …….	Won't need to be changed for this mission
takeoff_start_pos=-450,0 ……..	Change this to **takeoff_start_pos=0,0**
takeoff_stop_pos=1600,200	
landing_start_pos=5000,350	
landing_touchdown_pos=550,0	
landing_stop_pos=-250,0	

Be sure to change the **takeoff_start_pos** from **-450** to **0**. This represents your take off starting position relative to the longitude and latitude entered just above it. Usually this is the center of the runway but some designers use a different reference point. The **-450** means that you will start 450 yards back from the position entered. That's fine when measuring from the center of a stock runway, but in our case we entered the exact position from which we wanted to start. By changing the -450 to a 0 you will start at the exact longitude and latitude that you entered. If you failed to change this number you would start 450 yards farther back, behind the hangers instead of sitting in front of them.

Make the changes shown then save the mission. Start up CFS2, load the mission and go flying. When the game starts you'll find yourself parked in front of the hangers ready to taxi to the runway.

The view from the cockpit prior to engine start.

An outside view just after the engine started (note exhaust smoke)

That's all there is to it. Determine the exact take off longitude, latitude, heading and elevation that you want to use; create, edit and save your mission; open the mission file and change the parameters. **You must designate waypoint #1 as TAKE OFF when writing the mission.** It doesn't matter what base you take off from because you are going to change the exact location anyway. But you must write the mission so as to take off from someplace so that Mission Builder will enter the proper coding into the mission file.

It's just that simple. In fact it takes longer to talk about it then to do it. After you've modified one or two missions you'll find it second nature.

The technique works equally well for AI planes. Again, you must designate their first waypoint as **Take Off**, and then enter in the correct coordinates as we did above and they will depart from the new location. Look at the illustration on the next page.

Here you see a flight of Wildcats taking off from Peale Island in the Wake Island group. You can see that *Agate–2* is just getting airborne (notice the plane's shadow on the runway). The plane behind him is still on the ground. In the background you can see the main Wake Island airbase. The coral runway the Agate flight is using is infrastructure added through Mission Builder. To create the mission I placed the runway then zoomed in to get the exact take off coordinates and heading. Next I wrote the mission with the player airborne and the Agate flight **taking off from Wake**. Remember, you must designate the flight you want to change as taking off so that Mission Builder can insert the proper code. I had them "take off" from Wake because that was the nearest airbase. When the mission was finished I saved it, opened it in Notepad and changed the coordinates to match the location on Peale Island. The net effect is seeing an AI flight of Wildcats take off from an infrastructure runway that exists only in this one mission.

Here's another example of how to use this method. Say you want to have a seaplane take off from the open ocean as you approach. For this location I've chosen The Slot, northwest of Guadalcanal.

Here I've laid out the basic mission. I'm flying north in a Rufe and will encounter a Mavis about to take off. Notice that I have the first waypoint for the Mavis as **Turn** instead of take off. I've done that because the nearest base is miles away and setting the first waypoint to **Take Off** at this point would look confusing.

Once I have placed all of the waypoints and zoomed in to get the exact spot on the ocean I want the Mavis to take off from, then I set her waypoint number one as **Take Off**. As soon as I do that the airplane icon jumps to the nearest airbase, Rakata Bay on Santa Isabel Island in this case. Notice in the illustration below how far this is from where I want the actual take off to occur. You can see why setting the first waypoint to Take Off too early would be confusing.

Remember, I don't care where the Mavis icon jumps to since I am going to change that location when I edit the file. I just need it to jump to a valid take off location so that Mission Builder will write the take off code into the mission. With the mission saved I open it to do my editing. The original entry is on the left and my modified version is on the right.

```
[runway.0]
id=9140
heading=0
units_across=2
base_lat=S07* 35.31'
base_lon=E158* 42.97'
base_alt=+12
units_across=2
takeoff_start_pos=-500,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=500,0
landing_stop_pos=-250,0
```

```
[runway.0]
id=9140
heading=308
units_across=2
base_lat=S8* 8.98'
base_lon=E158* 30.03'
base_alt=+0
units_across=2
takeoff_start_pos=0,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=500,0
landing_stop_pos=-250,0
```

Notice that in this example it was necessary to change the altitude from 12 feet to sea level. Here's the result:

Mavis sitting on the ocean

Just lifting off (note shadow)

Headed out

Wingmen Issues As you saw above with the Peale Island example, you can use this technique for either single planes or flights of two or more. The same holds true for the player's flight – up to a point. If you create a take off point out in a vacant field somewhere you could have seven wingmen and it shouldn't cause any problems. However, under some circumstances additional wingmen could result in disaster.

Take the Wake Island mission as an example. When you add wingmen to your flight they line up in a set pattern behind your plane. Since your wingmen key on you, they use the same starting point that you do. If you added 4 wingmen to the Wake mission they would not line up on the runway, they would line up behind you. Some of them would be inside of or behind the hangers. When you start to move forward they also move, only they will run right into the hangers and explode.

Even if they didn't run into anything, as you taxi to the right and down to the end of the runway your wingmen will not follow. They are programmed to take off in a straight line from the starting point. While you are taxiing to the runway, they are taking off straight ahead, at right angles to the runway or running across it and colliding with the buildings on the other side before they get airborne. The only time I've seen wingmen not take off in a straight line is when using add-on planes that have high torque causing the plane to pull to one direction or another. Then they are pulled in the direction of the torque and take off at an angle. Review your new take off location carefully before adding wingmen to your flight.

Multiple Take Offs – Same Airbase I've included a sample mission titled *Henderson Alternate Take Off* with this Handbook that adds a new wrinkle to changing Take Off points. In this mission you are based at Henderson when it comes under surprise attack. Your plane is parked in one of the aircraft revetments. You must start your engine, taxi to the runway, take off and engage the enemy. As the mission begins another flight is already on the runway ready to take off. You will follow them.

On the surface the mission looks simple enough. The challenge comes from having one set of planes (AI) take off from the runway default location and another plane (yours) take off from a different spot adjacent to that same runway. Here's the solution.

Begin your mission by selecting Henderson as your takeoff location. Zoom in on the base and select one of the aircraft revetments near the runway. I chose the cluster near the western edge of the

airstrip. Center your cursor and note the longitude and latitude then drag the cursor and record the heading as in this illustration:

Note that the longitude and latitude shown in the illustration is that of the cursor's location after I dragged it forward. The center of the revetment (your take off point) is at **S9* 25.87' E160* 2.59'**.

Now go ahead and plot out the remainder of the mission. Lay in the waypoints for the other flight of Corsairs (or whatever plane you'd like) and those for the attacking Japanese, add any infrastructure you want, write up your triggers and events and test fly the mission. **DO NOT** set your first waypoint to **Take Off** until all of your testing is completed. Place your aircraft right above the runway and set the action as **Turn**. If you set your action to Take Off you will collide with the AI planes that are set to take off when the mission starts. Since you haven't changed your starting location in the mission yet, both of you will try to occupy the same place at the same time – that doesn't work. For testing purposes starting in the air will work just fine. Only after you have the mission tweaked the way you want it do you set your first waypoint to **Take Off.** Then save the mission and exit Mission Builder.

Open the **mis** file and scroll down to the runway entry. Normally you would simply change the longitude, latitude and heading and save the mission but that won't work in this case. You want the AI flight to take off from the runway and you want your plane to start from within a revetment. The

mission contains only one runway entry. If you change the runway entry to match the revetment location both your plane and the AI planes will line up on that spot. So, we need to create another runway entry just for your plane. **When you have multiple flights taking off from the same airbase but using different locations you need a separate runway entry for each flight**.

Begin by copying the entry for the Henderson runway and pasting it directly below the original:

[runway.0]	Original entry, nothing to change
id=8030	
heading=66	
units_across=2	
base_lat=S9 25.87	
base_lon=E160 2.85	
base_alt=55	
takeoff_start_pos=-450,0	
takeoff_stop_pos=1600,200	
landing_start_pos=5000,350	
landing_touchdown_pos=550,0	
landing_stop_pos=-250,0	
[runway.0]	Change this to **[runway.1]**
id=**8030**	**Change this as defined in the text**
heading=66	Change this to **154**
units_across=2	
base_lat=S9 25.87	Coincidently this doesn't need to be changed
base_lon=E160 2.85	Change this to **E160 2.59**
base_alt=55	
takeoff_start_pos=-450,0	Change this to **takeoff_start_pos=0,0**
takeoff_stop_pos=1600,200	
landing_start_pos=5000,350	
landing_touchdown_pos=550,0	
landing_stop_pos=-250,0	

The key line that needs to be changed to make the mission work is **id=8030**. ID 8030 belongs to the stock location for Henderson; we can't have two runways at different locations with the same ID number in the same mission so you must change this number to something different. Here's why you must change the ID number. Scroll up in the mission file until you find the Player's Formation information and locate the section **[waypoint_path.1]**. It will look like this:

```
[waypoint_path.1]
id=5000
waypoint.0=2,1,S9* 25.87',E160* 2.85',+0,,,,0,,8030,0,1
waypoint.1=1,0,S9* 25.56',E160* 0.26',+602,,,,184,,0,0,1
waypoint.2=1,0,S9* 24.74',E160* 4.13',+602,,,,184,,0,0,1
waypoint.3=4,1,S9* 25.87',E160* 2.85',+0,,,,0,,8030,0,1
```

Notice that the runway ID of 8030 appears in the line for both the first and last waypoint. The entry on the first line is the **Take Off** runway and the entry on the last line is the **Land at Base** runway. In missions where the first and/or last waypoints are designated as **Turn**, airspeed, rather than runway numbers will appear instead. Although the first line also contains longitude and latitude information, the runway ID number takes precedence in determining the take off point. In plain English this line means: *At waypoint number one the player will take off from runway number 8030*

which is defined below. As we saw above, runway 8030 is defined as the stock location for Henderson. For our mission to work we need to change the ID number in the Player's Formation section to match the ID number in the runway entry section for our new Take Off point.

Waypoint.0=2,1,S9* 25.87',E160* 2.85',+0,,,,0,,**8030**,0,1

These two numbers **must** match

[runway.#]
id=**8030**
heading=66
units_across=2
base_lat=S9 25.87
base_lon=E160 2.85

The runway ID number of the new Take Off point MUST match the runway ID number shown in the Player's Formation for the first waypoint.

With that determined, what number do you use? You can use any 4-digit number. The number you choose can match another runway number in your airbases.dat file that isn't used in this mission, such as 8015 (Wake), or it can be any other number, such as 9999. The key is that the two match. For this mission I chose to use the Wake ID of 8015. With the changes made the new mission file entries will look like this:

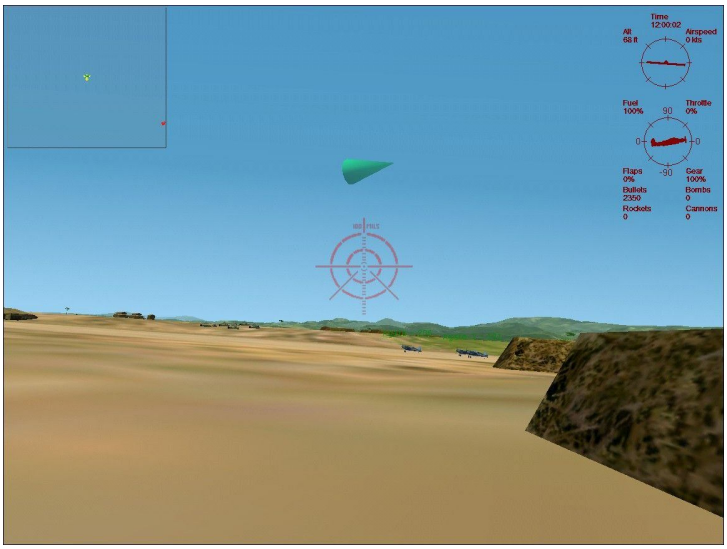
(**This is the player's formation info)**

```
[waypoint_path.1]
id=5000
waypoint.0=2,1,S9* 25.87',E160* 2.85',+0,,,,0,,8015,0,1
waypoint.1=1,0,S9* 25.56',E160* 0.26',+602,,,,184,,0,0,1
waypoint.2=1,0,S9* 24.74',E160* 4.13',+602,,,,184,,0,0,1
waypoint.3=4,1,S9* 25.87',E160* 2.85',+0,,,,0,,8030,0,1
```

(**This is the new take off location info)**

```
[runway.1]
id=8015
heading=154
units_across=2
base_lat=S9 25.87
base_lon=E160 2.59

base_alt=55
takeoff_start_pos=0,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=550,0
landing_stop_pos=-250,0
```

With the changes made, save the mission and go flying. When the mission starts this is what you'll see:



View from Corsair cockpit (note other flight)

View from behind Corsair

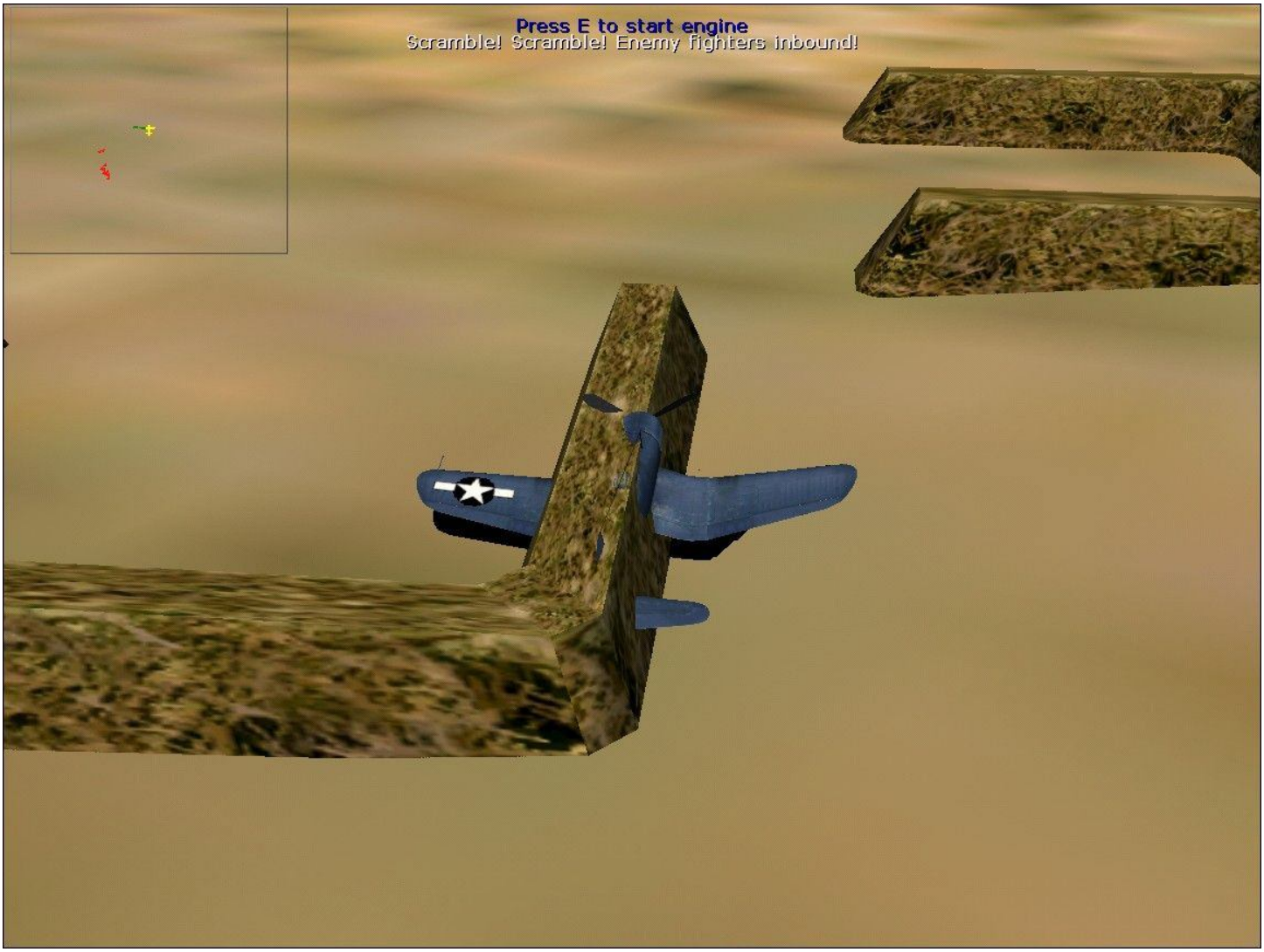
Corsair sitting in the revetment

Creating multiple flights using different take off points from the same airbase takes a little more effort but actually isn't difficult. Once you've done it a couple of times you'll find it quite easy to make the modifications. Remember these two rules and you won't have any problems:

1. **When you have multiple flights taking off from the same airbase but using different locations you need a separate runway entry for each different take off location.**

2. **The runway ID number of the new Take Off point MUST match the runway ID number shown in that flight's formation for the first waypoint (the player's formation if you are the one not starting on the runway).**

Pretty simple actually; but what happens if you make all of the changes described above and when you start the mission it looks like this:

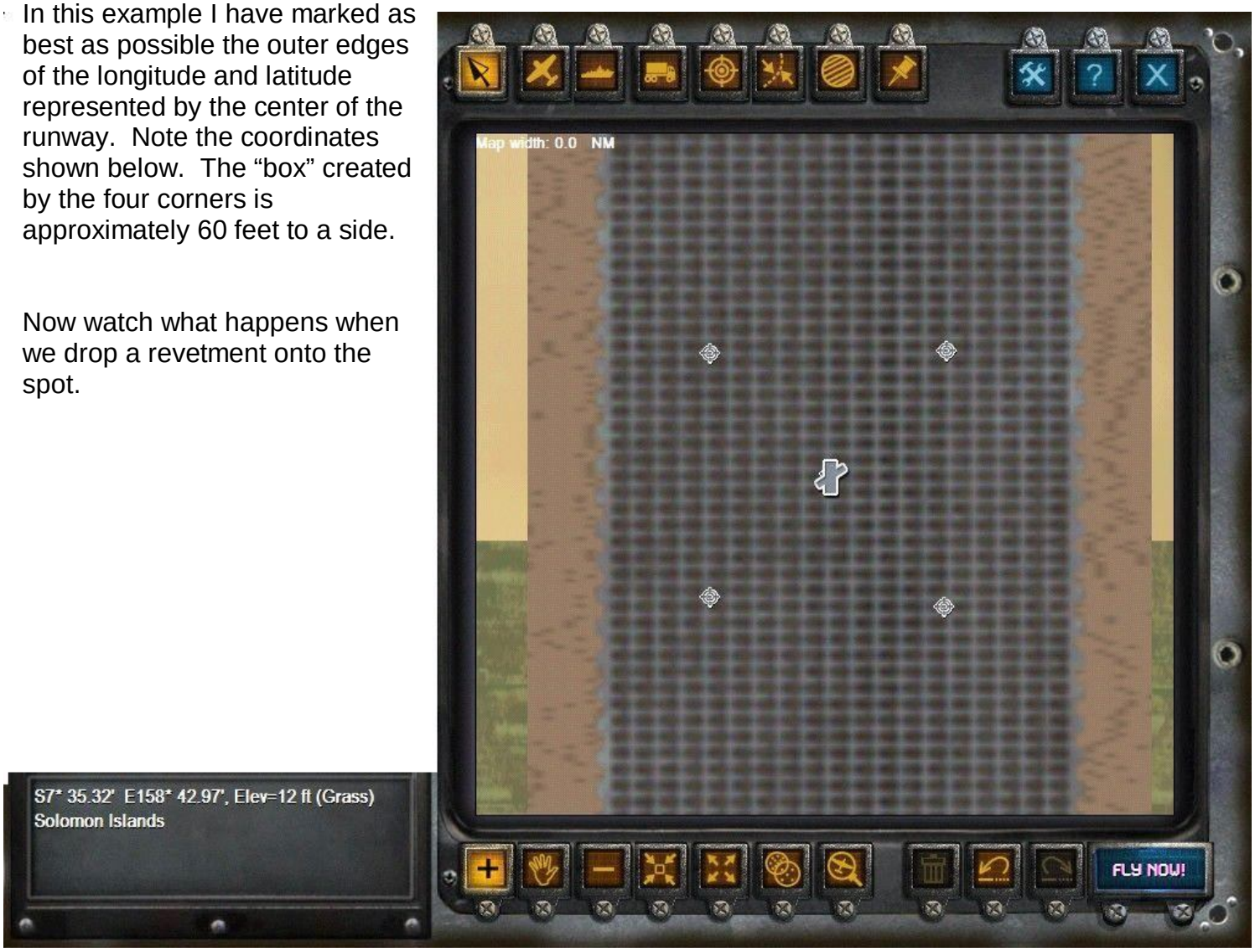


Oops! Read on for the explanation…

No Free Lunch and Other Trade Offs There is no free lunch in life and that definitely applies to CFS2. Although the use of longitude and latitude gives us a good location, it is not precise. In CFS2 longitude and latitude are carried out to only two decimal places. This results in any given coordinate being accurate to within approximately 10 meters. Picture a box a little over 60 feet on a side. If your cursor is placed anywhere within that box you will read the same longitude and latitude even as you drag the cursor around. This creates a pretty large margin of error when trying to match your plane's location to an infrastructure location. See the following illustrations.

In this example I have marked as best as possible the outer edges of the longitude and latitude represented by the center of the runway. Note the coordinates shown below. The "box" created by the four corners is approximately 60 feet to a side.

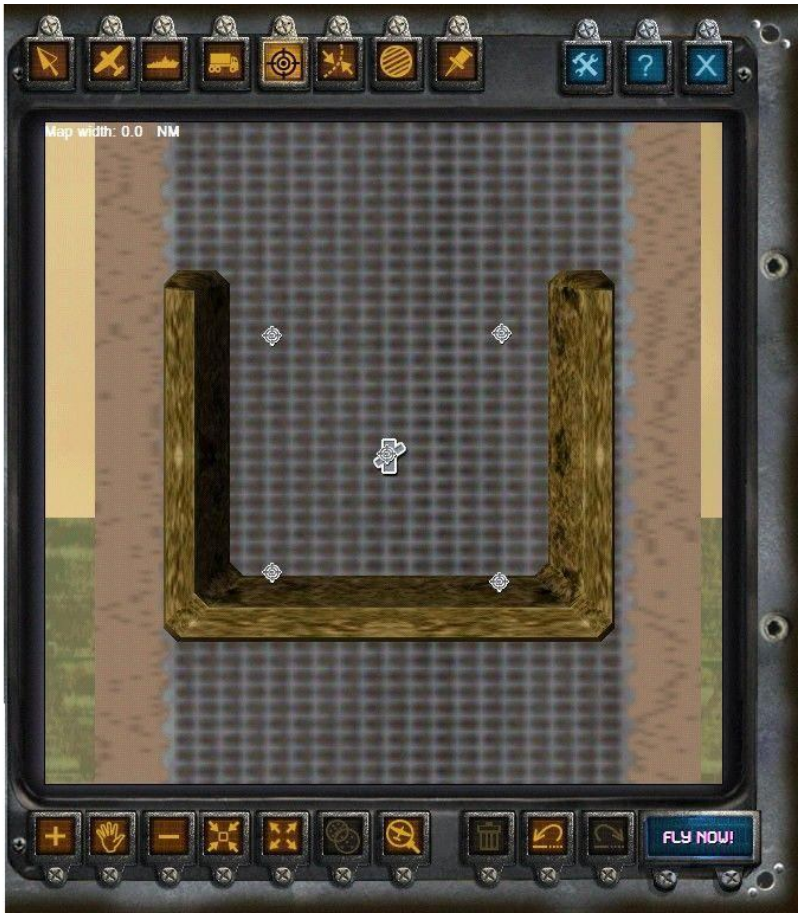
Now watch what happens when we drop a revetment onto the spot.



Here I have centered the revetment icon on the middle of the runway.

Notice that the revetment is slightly larger than the coordinate box, meaning that the edges of the revetment would have different coordinates than the center. However, as far as the game is concerned both the revetment and the runway have the same coordinates because the revetment icon is inside the “box”.

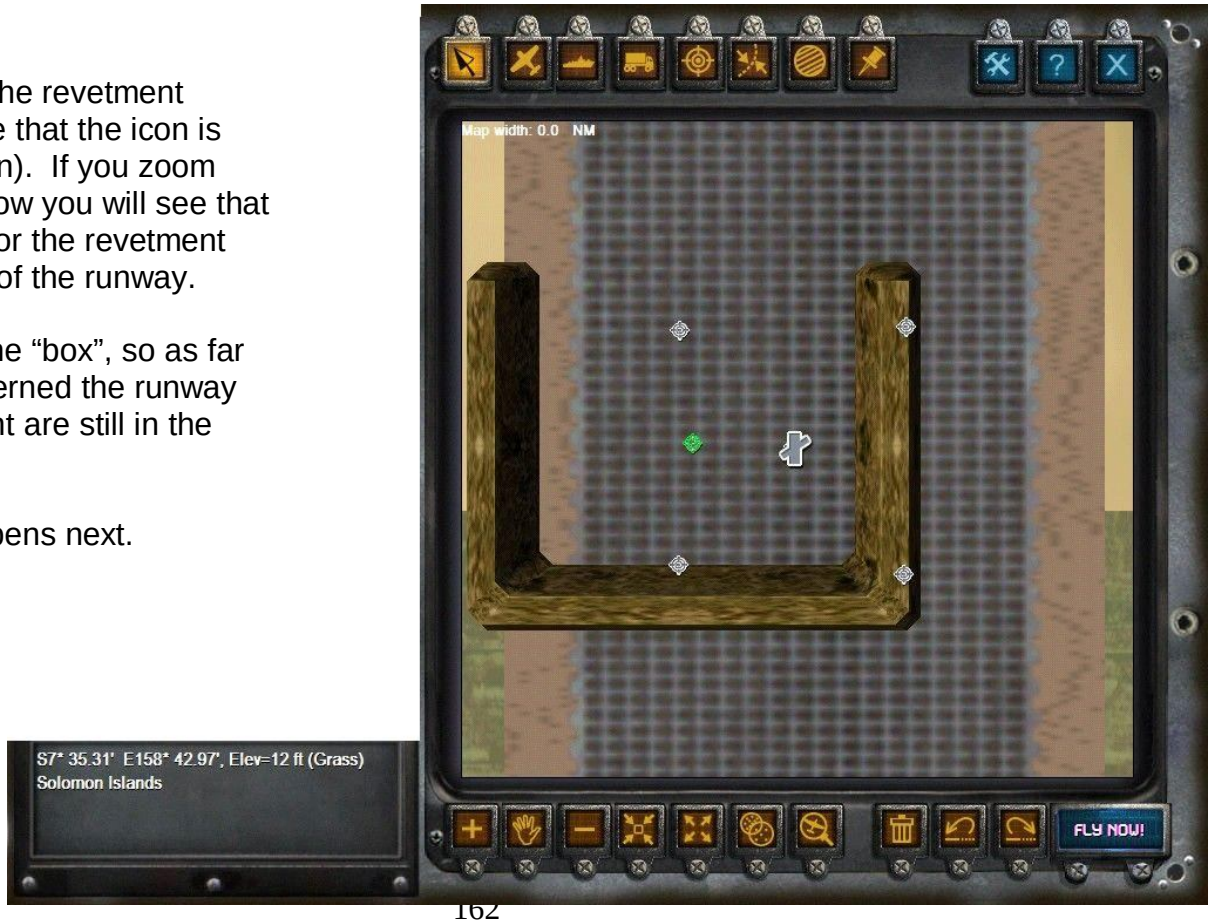
Watch what happens when I move the revetment icon.



Here I have slid the revetment to the left. Notice that the icon is highlighted (green). If you zoom in on the text below you will see that the coordinates for the revetment still match those of the runway.

We haven’t left the “box”, so as far as CFS2 is concerned the runway and the revetment are still in the <u>same</u> location.

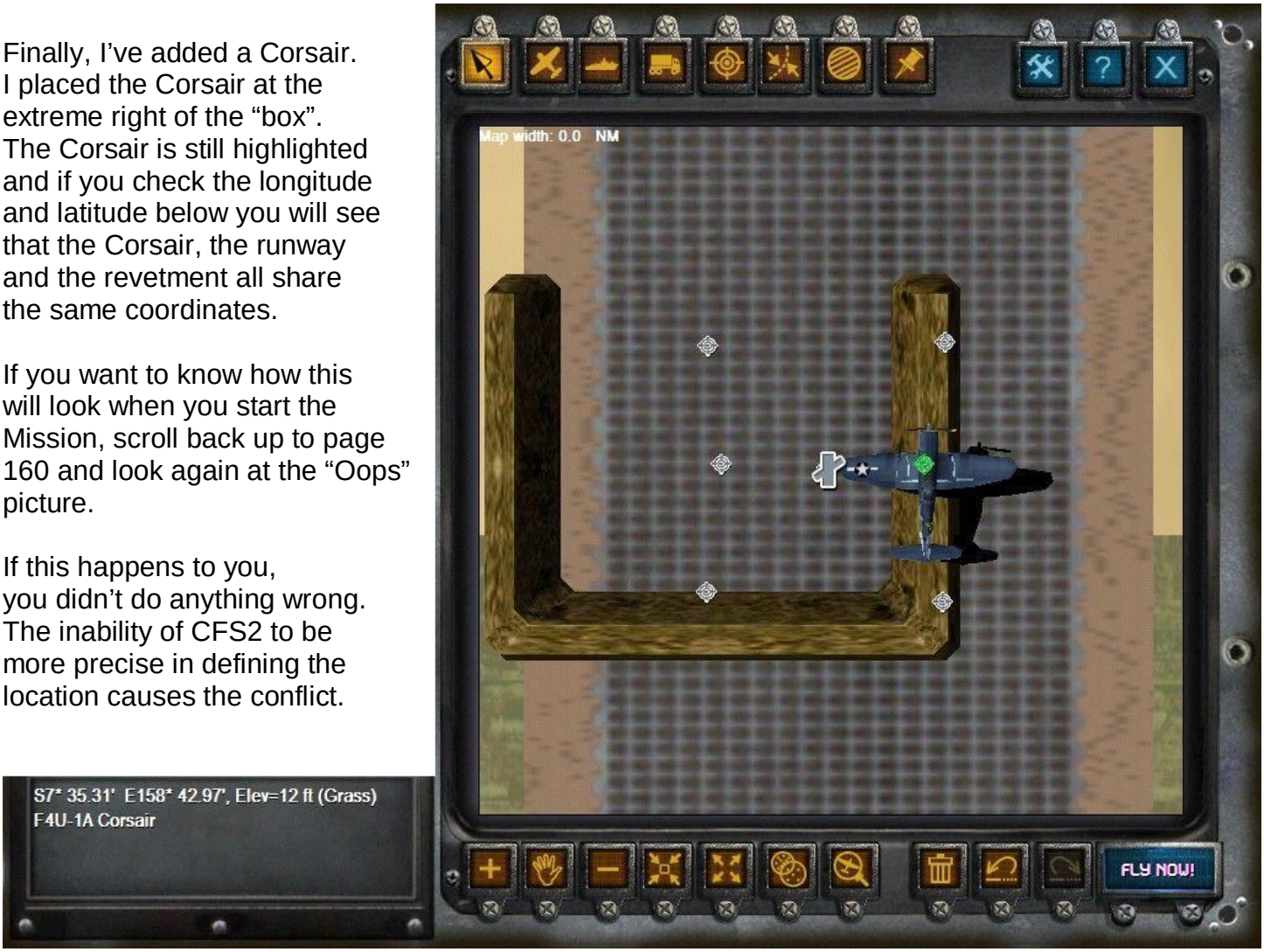
Watch what happens next.



Finally, I've added a Corsair. I placed the Corsair at the extreme right of the "box". The Corsair is still highlighted and if you check the longitude and latitude below you will see that the Corsair, the runway and the revetment all share the same coordinates.

If you want to know how this will look when you start the Mission, scroll back up to page 160 and look again at the "Oops" picture.

If this happens to you, you didn't do anything wrong. The inability of CFS2 to be more precise in defining the location causes the conflict.



This isn't a problem when taking off from an infrastructure runway that you added to a mission or when taking off from an open field. Also, it seems to be less of a problem when trying to match your take off location to that of infrastructure that you placed yourself versus infrastructure that is a part of the Global Scenery Layer or an add-on scenery file. Just be aware of it and recognize that you may have to do some trial and error until you get a workable solution.

Although it appears to me that CFS2 generally tries to place you near the center of the coordinates that you define, I have noticed some tendency to wander a bit. Meaning your actual location on the ground could vary slightly from mission to mission. In most cases I doubt that the variation would be enough to cause problems and may not even be noticeable to you.

Landing Points If you design a mission like the Wake Island example we used earlier, where you had only one flight leaving from the runway and changed its starting position, you are also changing its landing position.

As you know, if the final waypoint action is set to **Land at Base** then the final waypoint places you about 6 miles out from the airbase lined up with the runway. In the Wake Island example we redefined the location of that runway and changed its heading. Instead of lining up on the runway you will be lined up with the take off point in front of the hangers.

This isn't necessarily a problem as you can either press "X" and end the mission or you can turn and line up with either end of the runway and land as you normally would. In missions where you take off from one base and land at a different base this doesn't occur and your landing pattern isn't effected.

There is a solution to this problem that allows you to take off from in front of the hangers yet return lined up on the stock runway. It involves making a second runway entry in the mission file. I won't go into detail, but basically you would create a new runway entry matching the stock runway information, assign it a new ID number, and match that ID number to the ID number shown in the player's final waypoint. If you familiarize yourself with the steps in the *Multiple Take Offs – Same Runway* example I'm sure that you can figure it out. If not, email me.

Editing Issues There are two other issues that you should be aware of. If during the mission you go to Advanced Information, one of two things will happen; either the Advanced Info screen will appear as it normally does and you can continue to fly the mission or the game will Crash to Desktop (CTD).

Second, if you open the mission using Mission Builder and edit any facet of it one of two things will happen; either Mission Builder will CTD at some point or the mission will edit and save normally but you will find yourself at the wrong starting position.

Both of these issues share the same cause, the runway ID number that you choose to use for your new take off point. Whenever you use either the **Take Off** or **Land at Base** commands in Mission Builder the game accesses the airbases.dat file and copies the necessary information from that file and writes it into the mission. It always overwrites the existing information in the mission file. If the runway ID number in your edited mission doesn't exist in the airbases.dat file, Mission Builder CTD's.

You will recall that in the Wake Island example we did not change the runway ID number. It wasn't necessary since ours was the only flight leaving from that runway. While we did change the take off coordinates, we didn't change the runway ID number. When you edit this type of mission in Mission Builder the game will overwrite your entry with the correct coordinates for runway 8015, placing you back at the end of the runway. You will need to manually re-edit the file to return to the hangers.

In the Henderson example we created a new runway entry. You'll recall that I said you could use any 4-digit number for this new runway entry. The number you select will be one of two things; it will either match another runway entry already in the airbases.dat file but not used in this mission or it won't. The number you use will determine what happens when you edit the mission in Mission Builder.

- If the second runway ID number matches an ID number for another runway in the airbases.dat file (as in our example where we used 8015, Wake's ID), when you save the mission, Mission Builder will place you at that runway (you'd find yourself starting the Henderson mission at Wake).
- If the second runway ID number doesn't match an existing ID number in the airbases.dat file (for example, using 9999), the Mission Builder will CTD during the editing process.

Something similar happens when you try to access Advanced Information in a mission. If the game finds the runway ID numbers that you used you will see the Advanced Info screen and the game will play normally. If you used a non-existent ID number the game will CTD.

None of this is fatal; it is merely something to be aware of. That is why I emphasize to do all of your tweaking prior to setting the new take off point in the mission file. The two key things to remember are:

- If you want to edit the mission in Mission Builder then the runway ID numbers in your mission must match those in your airbases.dat file, **even if you change the exact locations.**
- Editing the mission in Mission Builder resets the take off position to the original coordinates for the runway ID number used, requiring you to reenter them via hand-editing in the mission file.

If you write a great mission and never have to edit it, this will never be an issue for you. Just be aware that it can happen and what causes it.

Don't be intimidated by this technique. It may sound difficult but it really isn't. Once you have done it a couple of times it will become second nature. Just remember the key points outlined above. Remember, this method should not be used for repeated take offs from the same location. If you anticipate using the same take off point over and over again then you should make an entry into the airbases.dat file so the new location will show up in the Mission Builder and you can select it easily. The method taught here is perfect for altering an individual mission or for taking off from an infrastructure runway that you've added to the mission. It offers yet one more way to enhance the CFS2 experience through creative mission building.

Quick & Dirty – But Not the Best Solution

Player's are always looking for a shortcut or a quick solution to doing things in CFS2. Here's a quick and dirty trick you can try for creating a new take off spot. It works best with seaplanes taking off from the ocean but even then, you may get crashes (of your plane not the dreaded CTD). I'm passing it along since some of you will discover it anyway and you should be aware that while it might work some of the time, it isn't as reliable as the method described above, which always works.

Setup your mission as you normally would with your seaplane located at the spot where you want to take off. Leave your waypoint #1 action as **Turn** instead of Take Off. Save the mission then open it in Notepad. Scroll down to the player's formation and locate the waypoint data. In the sample shown here I am just using two waypoints, you'll probably have more.

```
[waypoint_path.0]
id=5000
waypoint.0=1,0,S7* 48.29',E158* 39.28',+367,,,,112,,0,0,1
waypoint.1=4096,0,S7* 42.50',E158* 30.00',+602,,,,147,,0,0,1
```

On the first line note the numbers **+367** and **112**. This is your starting altitude (367 feet) and speed (112 knots). Simply change these two numbers to a zero, **0**. Now you'll be sitting at zero altitude (sea level) with your engine off. That's it. Quick and dirty, as I said. If you have wingmen you will need to make an additional edit. Look above the waypoint section at the player's formation section (I am showing only a portion of it here).

```
unit_id_position.0=9155,0,0.0,0.0,0.0
unit_id_position.1=9156,0,-300.0,-600.0,0.0
unit_id_position.2=9157,0,300.0,-600.0,150.0
unit_id_position.3=9158,0,600.0,-1200.0,150.0
unit_id_position.4=9159,0,1200.0,-1200.0,300.0
```

Note the numbers **150** and **300** near the far right for units 2, 3 and 4. This is their altitude above the player when the mission starts. You must change these to zeros or these planes will fall into the ocean and explode when the mission starts.

This sure seems easy, so why don't I recommend it? CFS2 does not come with any seaplanes, so any seaplane you use must be an add-on. You will find from time to time that the contact points on some of the add-on aircraft aren't set quite right, causing the plane to be sunk slightly into the ground or hover slightly above it. If your contact points aren't perfect, when you use this edit you may find your plane sinking, crashing into the sea, exploding, or you get a "Prop Strike" message and the engine won't start. I don't get those problems when following the more detailed steps outlined in this chapter. Also, this method doesn't write the take off code into the mission so AI flights behave oddly; some take off, while others just sit there. It isn't a reliable method to use.

Still, some of you may want to try it. If it works for you – great; if not, don't say I didn't warn you!

Chapter sixteen – error messages

Preflight
If you design enough missions sooner or later you will end up with an error message. In fact, I'd go so far as to say that if you haven't gotten any error messages to date you are still a mission building novice. Have no fear, while they may be frustrating, they are usually not fatal.

In this chapter I will try to go over some of the more common error messages and what can cause them. This is not the definitive, all-encompassing solution to mission building errors. In many cases they are multiple problems that can cause the same error message and I'm sure I won't have all of the answers here. But drawing on my experience, I can give you some good places to look first and hopefully save you hours of frustration.

Avoiding Errors in the First Place
Obviously the best way to solve your mission building errors would be to avoid getting them in the first place. Probably the only way you could do that is if you had a purely stock CFS2 installation and never added or changed anything, but that wouldn't be much fun. Plus, I'm not sure even that would guarantee an error free life. One of the real advantages of CFS2 is being able to build on the basic simulator by adding all sorts of custom planes, ships, vehicles, infrastructure, scenery, etc and tweaking the game to meet your particular personality. A basic stock installation is a little less than a gigabyte in size. Once you start taking advantage of all of the add-ons out there this can rapidly grow to 3GB, 6GB, 10GB or more! Unfortunately, those wonderful extras that add so much to our CFS2 experience can also bring headaches in the form of errors and crashes with them.

The number of available downloads for CFS2 is incalculable, and most of it is free for the taking. Some of these add-ons are of first rate quality, better even than anything Microsoft produced. They are well researched, thought out, carefully designed, tested and debugged to give you years of trouble free use. Others were pulled together haphazardly by well-meaning enthusiasts whose intentions were good but who failed to put the necessary time and effort into producing a trouble free product. These downloads are headaches waiting to crash your game or give you mission building fits. Unfortunately there is no real way to know in advance what you are getting. Even with all the problems a bad add-on can cause, I still encourage everyone to load up on the freebees, keeping the good ones and dumping the rest, because it is through expanding and building on the basic installation that we can derive the most pleasure from this terrific flight simulator.

Here are a few steps you can take to reduce problems and errors in mission building and elsewhere:

- **Keep your PC tuned up** – This applies to every facet of home computing. Have a good, up to date, anti-virus program and firewall installed, and use them. Rid your system of Spyware, Adware, and other accumulated junk. Defragment your hard drive as necessary. Keep your registry cleaned out and avoid loading up your System Tray with a lot of processes that slow your system down. There are many utilities out there for maintaining your PC in tip-top shape. If you're not much of a geek, then consider investing in one or more of the suites designed to keep your system running smoothly. If your computer if running right and bug free, your CFS2 world will run smoother as well.
- **Do Regular CFS2 Backups** – This goes without saying yet I still read on forums where a player suffered a hard drive crash and lost everything they had entered into CFS2. Regularly scheduled backups can really save the day for you if you ever do lose a drive or have a total Operating System failure. I backup certain CFS2 text, data, and configuration files daily, and

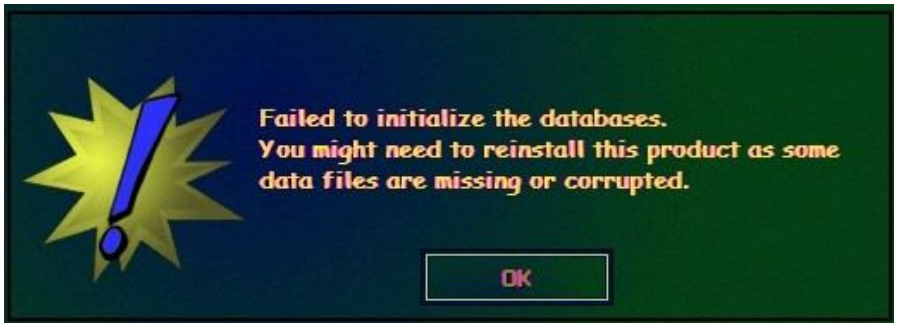
do a complete backup of each of my installations weekly. If (or rather when) something goes wrong and I have to reconstruct an installation I have good files backed up that I can easily copy over into my new install. It gives me peace of mind and quickly gets me back up and flying (As an example, I had to do one reinstallation while writing this Handbook!).

- **Regularly Save Your Missions** – There is nothing worse then spending an hour or two working on a mission only to have the Mission Builder CTD before you're finished and have saved the mission. Avoid this hassle by regularly saving your mission along the way. On big missions I save after plotting all of the waypoints, after entering the triggers, after entering the events (I frequently have my missions planned out in advance and as a result do all of my triggers at one time followed by all of my events). Remember, never overwrite a mission; always save each new version using a different file name. Overwriting missions can cause big problems.
- **Be Selective in your Downloads** – Do you really need every Spitfire that's available for download? Learn to be selective in your add-ons. Test each one, keep the best and discard the others. If you shop around you can almost always find a good quality version of whatever it is you are looking for. Try to keep your installation the best of the best.
- **Test as you Install** – Each time you add something to CFS2 you should test it to see if it works properly. Test the plane, object, scenery, etc. both in the game and in Mission Builder. This really helps to find problem add-ons early, before they have a chance to "disappear" in your folders somewhere. I recommend testing after each add-on rather than once a day or once a week. I know, it's slower, but it's safer too.
- **Document your Add-Ons** – You might consider keeping a log of everything you add to your game. That way if something pops up suddenly you can track what was most recently entered and begin backing things out in reverse order to see if one of your recent add-ons is at fault. This is like a fail safe to testing after each addition.
- **Work Carefully and in a Logical Manner** – CFS2 might be a game but at its heart it is still a computer program that runs on complicated computer code. The Mission Builder is basically a graphical interface for writing some of that code. As long as you stick to using Mission Builder you're pretty safe. But once you start hand-editing missions the chance for errors really jumps up. Take your time, think through your edits and double check your work. Computers aren't very forgiving. Misplace a comma here or a period there and CFS2 can no longer read your "code" and the mission crashes. Hand in hand with that goes…
- **Back up your Original Files First** – Before hand-editing a stock mission, or any mission that works fine as it is, always make a back up copy first. That way if things go terribly wrong you still have an unblemished copy to reinstall or copy from and start over on. I also recommend that you never overwrite any stock file with an add-on without first saving a copy of the unaltered file in some safe place. Create a new folder in your main CFS2 folder named **BACKUPS** and store all of the original files that you replace there. That way you'll have them handy if you ever need to replace one.
- **Keep a Second Stock Install on your System** – If you have the room on your hard drive, consider having a second, purely stock, install of CFS2 loaded. Don't ever add anything to this install, just use it for testing how things are supposed to work. Also, it can come in handy when you need to replace a stock file in a hurry. A corrupted **cfs2.cfg** file can stop your game in its tracks. Having another one handy to drop in its place can get you back up and running in a snap.

Even when you take every precaution there is still the likelihood that something will go wrong and you'll get an error message in Mission Builder. Starting on the next page we'll look at some of the more common error messages you might see and try to identify where the problem lies.

Common Error Messages

The following are some of the more common error messages you might see and their possible causes. This is not a complete list nor does it have every possible cause listed. I know I don't have all of the answers and I am sure there are other things that can cause these errors that I am unaware of. All of the information I provide here is based on my own personal experience in writing thousands of missions for CFS2. If you positively identify some other cause of these problems please let me know and I'll add it to my ever growing list.



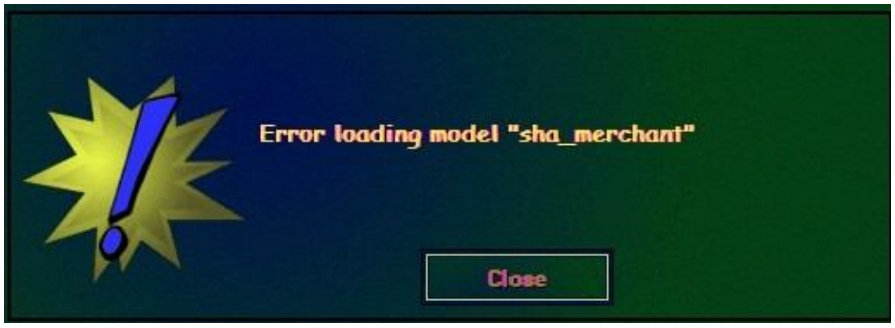
This is probably the most dreaded (and most common) of all CFS2 error messages. The first thing you need to know is DON'T PANIC! Reinstalling CFS2 is probably the last thing you need to do. The most common cause of this error message is too many planes in your AIRCRAFT folder. There is a 100 plane limit in CFS2 that prevents Mission Builder from loading. The game will still load, although you will see this error message when you try to fly Free Flight or Quick Combat. In the game however, you can click OK a couple of times and pass through this error. In Mission Builder you are stuck, it won't load.

The best way to avoid this error is to keep the total number of planes in your AIRCRAFT folder to fewer than 100. You can do this manually or you can download one of the utilities out there that makes the job easier. I recommend using Martin Wright's **ACMan**. It easily allows you to move aircraft in and out of the game and keeps a running total of the number of planes you have entered so you know how close you are to the limit. Keep in mind, if you have a single airplane folder with two textures, that counts as two planes toward the total limit of 100.

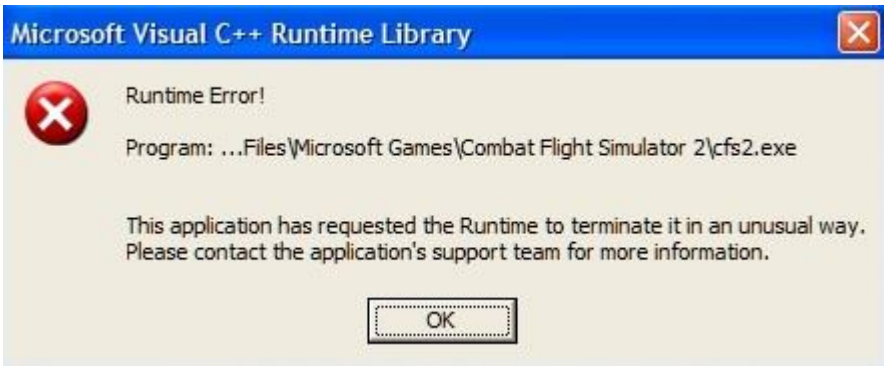
Some players report that you can also get this error message if you have too many ships. I have not experienced that and I have almost 200 ships in one of my installs. What I have seen is that a single bad ship, one with corrupted files, can cause this error; another good reason for testing each time you add something to the game. It is also possible to get this error when you have too many DP files in your OBJECTSDP folder. The limit is said to be between 1,800 and 1,900 DP files. I haven't been able to pin down an exact number and the more experience I have with CFS2, the more I believe that it is a combination of factors (too much of this, too many of those) that will cause this error.

If you are under the 100 plane limit and you don't have a bad ship or too many DP files, then you do have a problem. I'd begin by removing the latest add-ons you've placed in the game one at a time in reverse order and try starting Mission Builder after each one. That might pin point the problem file. If that doesn't work, then you probably have a corrupted file someplace in the game. If you recently modified a file anywhere in the game, try removing it and replacing it with your backup (you did backup before changing anything didn't you?). If that still doesn't work then you are probably looking at a reinstallation.

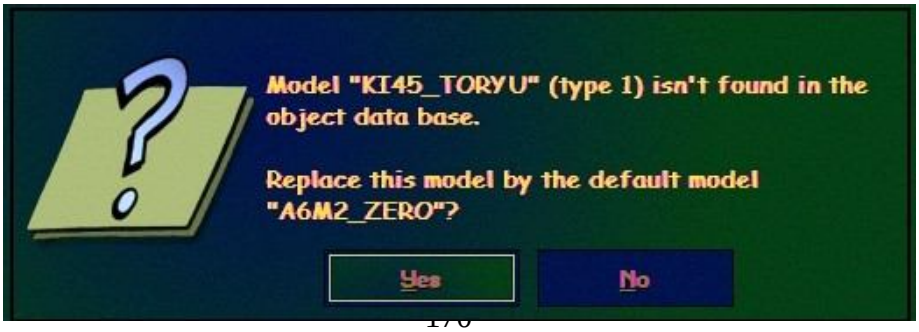
Reinstalling isn't as big a deal as it sounds. Start by renaming your existing CFS2 main folder to something else, for example **Combat Flight Simulator 2 – Pacific**. With the main game folder renamed you can now use your installation disks and install a second version of the game. The installation disks are looking for a folder named **Combat Flight Simulator 2.** By renaming this folder as shown above the install disks won't find it and you can install a clean, stock version of the game. Gradually copy your custom files and add-ons over to the new installation from the old one testing after each batch (for example after copying over the AIRCRAFT folder) to see if you can isolate the problem file. You may never find it. Sometimes things really do get corrupted, especially if you have CTD's during play, and the only solution is a reinstall. After you've done this a few times it becomes much less traumatic.



This error is caused by a bad file within the aircraft, ship, object, etc. that is shown in the error message. You'll need to fix that file or replace the object with something else.



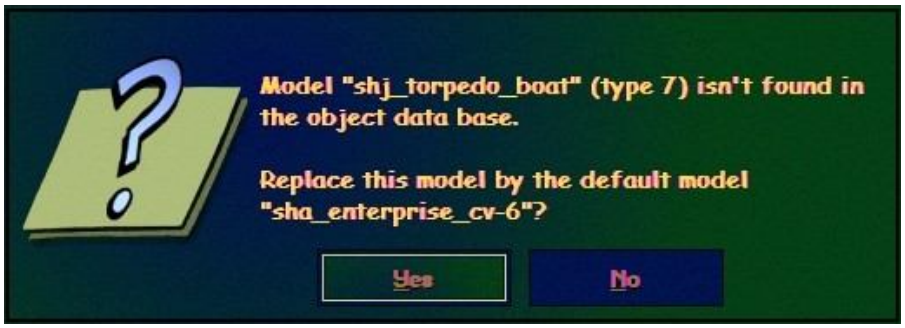
This error can be caused by a problem in your **airbases.dat** file. The error can be as small as a missing zero or misplaced comma and almost impossible to find. If you see this error, stop and think about what you were last modifying in CFS2. If you recently added new runways to your airbases.dat file then go there first and double check your work. Something isn't right in one of your entries. Try dropping your backup copy of the file into the game and see if it will start. If you still get this error then the problem could be almost anywhere.



You'll see the error message on the previous page any time that you try to load a mission that contains an object you don't have in your game. Mission builder will identify the object then ask if you want to replace it with the stock Zero. If you click **Yes** the mission continues to try loading, if you click **No** you will see this error message:

Click **OK** and the mission won't load and you go back to the main Mission Builder screen. Refer to the sidebar **Converting CFS1 Missions to CFS2** in Chapter Fourteen for more information.

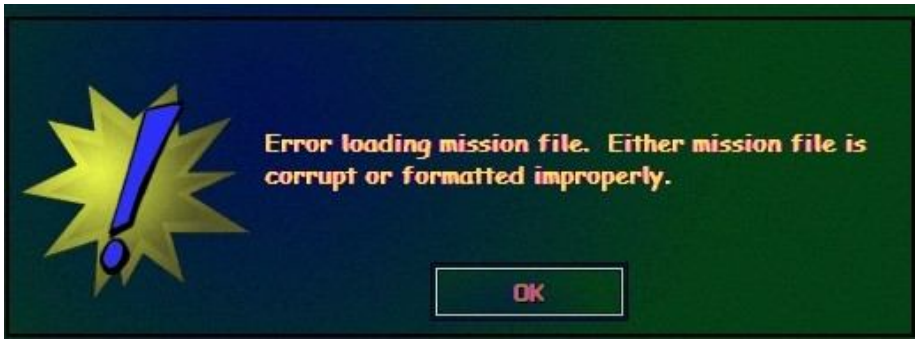
But what about when you do have the object in your game and still see this error message:



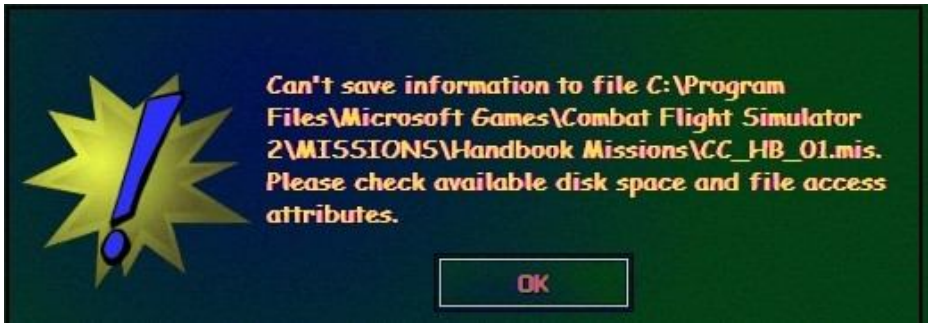
When you see this error and you know you have the object, the Japanese torpedo boat in this case, look carefully at the message. Notice where it says **(type 7)**. This is a reference to the **unit_family** shown in the mission file and in the object's DP file. If there is a mismatch between the two, when you try to open the mission in Mission Builder you will get this message.

Good luck contacting their support staff with a ten year old flight simulator problem! You'll see this error when your mission file is corrupted. This can easily occur when making a mistake during hand-editing. Again, work slowly, double check your work, and be logical in your thinking. Some minor

errors will be overlooked by Mission Builder entirely or result in the error message seen below. Others, such as messing up longitude and latitude, can result in this error.



If you see this error message after having hand-edited your mission then you did something wrong and you should go back and double check those edits. You might see this on a mission you downloaded from the Internet. If so, finding the problem can be tricky. The error could be almost anywhere, such as accidentally deleting the reference to one of the objects. Without knowing what was originally written into the mission, trying to find the error can be almost impossible. Still, check the file for obvious errors or missing sections. With a downloaded mission your best bet is to try downloading again or contacting the author.



You'll get this error message if you try to overwrite a file marked **Read Only**; another good reason to never overwrite a file but rather save each modification under a new file name.

Crash to Desktop (CTD) There are few things more aggravating than having Mission Builder CTD right when you were in the middle of writing a mission. The first step in diagnosing the problem is to remember exactly what you were doing when the crash occurred. That might help you isolate the problem area. Here are some of the more frequent causes.

One problem area is the **airbases.dat** file and its relationship to the Advanced Info screen and the runways you are using in your mission. At some point in your testing you hit **Advanced Info** and Mission Builder, or the game, CTD's. In my experience a CTD in connection with Advanced Info has always traced back to a conflict between the airbases.dat file and the airbase information stored in the mission file. We have talked about this problem above in Chapter Fifteen, *How to Take Off from any Spot on the Planet*, but a little recap won't hurt.

Open any mission where you takeoff or land and you will see a runway entry that looks similar to the ones in your airbases.dat file. The biggest difference is that the runway numbers in the mission file start at "0" and run consecutively. So in a mission that uses Henderson, the runway number may be "0" while Henderson might be runway number "103" in your airbases.dat file. That difference doesn't

matter. The key thing to remember about these runway entries in the mission file is that they are **not** copies of what's in your airbases.dat file, rather they are copies of what's in the author's airbases.dat file. If you didn't write the mission, that's a key distinction.

In the game, your starting position and waypoints are determined by the longitude and latitude that's entered into the mission file. The runway ID numbers (the 4-digit numbers) don't necessarily come into play. You don't have to have the same scenery installed in your game to fly a mission that I wrote in my game. That's a good thing. I could send you a mission I wrote that takes place in Borneo and even if you don't have any of the same airbases or scenery in your game, you could still fly it. You might take off and land in an empty field but as long as you have the same aircraft (folder names) in your game and any infrastructure that I added such as ships, buildings, etc. the mission will work. However, as soon as you hit Advanced Info the game will probably, but not always, CTD.

When you click on Advanced Info you are creating a link between the game and Mission Builder. The map you see in Advanced Info is the map from Mission Builder. When trying to match the airbases used in the mission to the map from Mission Builder the game looks to the airbases.dat file and not what is entered in the mission itself. What's important at this point is not whether or not you have my Borneo airbases entered in your airbases.dat file, it's whether or not you have the runway ID numbers assigned to my Borneo airbases in your airbases.dat file. If the game finds those ID numbers, even if you have them assigned to other airbases, you will see the mission map. If not – Crash to Desktop.

If you opened that same mission in Mission Builder you would see the mission map showing Borneo and all of the waypoints, etc. You wouldn't see any "X"s for the airbases unless you had the same Borneo entries in your airbases.dat file (we're assuming that you don't). You could even click *Fly Now* and fly the mission without any problems provided you didn't go to Advanced Info. However, when you try to edit the mission, CTD. The crash will either come when you first move an object or when you go to save the mission depending on exactly what you are editing, but either way it crashes.

Since there isn't one standard list for ID numbers for every runway in the world, designers can add their own. Sometimes add-on scenery has an airbases.dat entry that includes an ID number and sometimes they don't. As a result, except for the stock runways, we probably all have different runway ID numbers assigned to the various bases in our airbases.dat file. Even if you and I have the same add-on scenery in our games, if my ID number for the runway is different from yours it can result in a CTD.

There are many different combinations that can cause problems. For example: none of the ID's or geographic locations (longitude and latitude) in the mission match your airbases.dat file; some of the ID's/geo's match and some don't; the ID's match but the geographic locations don't (the designer and you used the same ID number for different bases); the ID numbers don't match but the geographic locations do (the mission starts at Darwin and ends at Ternate and both of you have the same scenery but you gave different ID's to those bases),etc., etc. In some cases the mission and Advanced Info will work if the takeoff data is correct even if the landing airbase information is different.

I could list all of the different combinations that I've encountered and the type of crash that they caused but you wouldn't gain anything from that. Suffice it to say that these conflicts between the mission runways and your airbases.dat file can cause:

1) no problem
2) a mission failed to load error message

3) a CTD when you start the mission
4) a CTD when you hit Advanced Info
5) a CTD when you try to edit the mission in Mission Builder

The bottom line is that any time you encounter a CTD when hitting the Advanced Info button or starting a mission, especially if you are flying an add-on mission, always open the mission file and check the airbase information first. In all likelihood, you have one of the conflicts described above.

Fixing the problem may not be easy. You could hand-edit all of the missing ID numbers to ones you have in your airbases.dat file that are assigned to other bases. That should allow you to view Advanced Info without crashing. However, if you subsequently edit the mission in Mission Builder you'll find yourself at the bases defined by your airbases.dat entries and not at the original locations. You would then have to go back in and hand-edit as described in Chapter Fifteen to correct the locations. You can see how a problem like this can keep snowballing. Ultimately, you'll have to decide if it's worth the effort just to access Advanced Info or edit the mission in Mission Builder.

You can also get a CTD on your own missions when you have duplicate runway ID's in your airbases.dat file. That can happen easily as you add airbases. Always do a search of the file for the ID number you are entering when adding a new base before you save the file. If the number you plan to add is already in use, select a different number. It will save you grief later.

NEVER OVERWRITE A MISSION. Overwriting a mission again and again can cause hidden problems that may cause the mission to CTD or not play out properly, with missed waypoints and all measure of goofy things. When modifying a mission, always save the newer version with a different name. It's easy to simply add a letter or number suffix to each new version. When you are finished and the mission is perfect, save the final version with the name you want and delete all of the prototypes.

Another thing that will cause Mission Builder to CTD is too much text in the mission briefing or debriefing. If you type in a lengthy debriefing and exceed the size limit, when you click **OK** and close the event Mission Builder will crash. If you want to add a long briefing or debriefing do this. Create the debriefing event and for the text enter **Place Text Here**. Now save the mission and exit Mission Builder. Type up your text in Notepad and copy it. Next open the mission **dyn** file and scroll down to where you see your "Place Text Here" entry. Simply paste the new text over the old, save the file and you're done. Once you've added extra text like this you can still open the mission in Mission Builder and edit it as long as you don't open the debriefing event. If you do it will CTD.

Some add-on effects can cause a CTD. One of my favorite effects is Nanni's Landing Lights. They are burning oil drums that you can place in your mission for night landings. You can even spawn them so they don't light up until you get close to your base. They are really effective; however they almost always cause Mission Builder to CTD for me. Sometimes Mission Builder will crash after the mission instead of returning to the map screen, other times it crashes at random while editing (they cause missions flown in the game to crash too). I can usually get 3 or 4 error free missions before this starts to happen. The obvious solution is to quit using them; but they are so realistic that I don't want to give them up. What I do is place some type of infrastructure other than the oil drums as a place holder in my mission, do my testing and editing, then when I am satisfied, I insert the oil drums last and save the mission. This doesn't eliminate CTD's in the game but it does in Mission Builder. If you encounter add-ons that cause CTD's you've got to decide; dump them or find a work-a-round.

Sometimes Mission Builder crashes for no apparent reason. We've been tinkering with CFS2 for over ten years now but no one has all of the answers. There are some things we just don't understand or haven't figured out yet. Occasionally Mission Builder will begin acting very strangely. I've seen it create alternate waypoints or change waypoints all by itself. When it starts behaving like this the only cure I've found is to exit, delete the problem mission and start all over again. I've seen repeated crashes that appear to be caused by an add-on plane, even when it's a plane I've used many times without problems in the past. CFS2 is a very complex program and when you're writing missions you are effectively writing computer code. Sometimes things just happen. When it keeps crashing for none of the readily identified reasons, try exiting, deleting the offending missions and starting over again from scratch. Considering the thousands of missions I've written, the total number of unexplained crashes I've had is actually pretty small.

Aircraft Not Showing Up in the Mission Builder Any plane that shows up in Free Flight should also show up in Mission Builder. After installing a new aircraft you should always check it in Free Flight first to make sure it is working properly. If it isn't, read the documentation that came with the plane and double check that it is complete and not missing any files, etc. I won't go into the steps for installing aircraft into CFS2 as the focus here is on Mission Building.

Only Player Flyable aircraft will show up as an option when you first start up Mission Builder. That is why, in a stock installation, planes like the B-24 and B-25 aren't on the list. They are AI aircraft. All aircraft, Player Flyable and AI will show up on the list of available planes for the AI flights. Refer to the sidebar **What's in a Name** in Chapter Thirteen for information on which aircraft name will appear in the drop-down list.

I have seen an odd glitch in Mission Builder. Sometimes a plane that will show up in Free Flight will not show up in Mission Builder when the aircraft's DP file is missing the **[MISC_DATA]** section. This is the section in the DP file that contains information on when the aircraft entered service, its allegiance, category, family, performance information, and other things. All stock aircraft have this entry but some designers do not put it in their add-on aircraft. If you have a plane that shows up in the game but not in Mission Builder look here first for the solution. I don't know why Mission Builder should require this data. In fact, I have some planes that don't have a [MISC_DATA] section and they show up just fine. Others are missing it and as soon as I enter one they show up, no problem. Just chalk this up to one of those oddball quirks of CFS2 and the Mission Builder. If you do need to enter a new [MISC_DATA] section you can simply copy one from a similar plane, being sure to copy a fighter's info into a fighter, etc. Be sure to delete the aircraft's **CDP** file after making changes to the DP file. If you don't, the changes won't take effect. Don't worry; the game will recreate a new CDP file as soon as you next use that aircraft.

Error Log A possible source of assistance when Mission Builder crashes or a mission won't load, either in Mission Builder or in the game is the Error Log. After a failure as described, go to your main CFS2 folder and look for the file named **error**. Open the file with Notepad and you'll see a brief explanation of the error. Here are a couple of samples:

[Mission file error]
error=Unable to init unit 1000 ……. **the nature of the error**
title=Bombing Berlin …….. **the mission name**
filename=C:\Program Files\Microsoft Games\CFS2 Europe\MISSIONS\ Bombing Berlin ... **File location**

[Mission file error]
error=Unit type ID 3800 not found (specified for unit 9082)
title=Raid on Yap
filename=C:\Program Files\Microsoft Games\Combat Flight Simulator 2\MISSIONS\Raid on Yap

Lines three and four give you the mission name and its location, two things you probably already knew. The second line is the important one. While it doesn't tell you in plain English what happened, it does tell you where to look.

The first example says "Unable to initialize unit 1000". If you open the **mis** file and search for unit 1000 you can identify the aircraft, ship, object, etc. that was the problem. Now you know where to look. If you can't find the trouble with the object then you can replace it with something else and try the mission again.

In the second example, if you open the **mis** file and locate unit 3800 you'll see the example shown to the right:

```
[unit.4]
id=9082
type_id=3800
is_player_aircraft=1
allegiance=4
skill=1
payload=0
```

You can see that the type_id of 3800, which is the cause for the error, is assigned to unit id 9082. You would now scroll down to the list of objects near the bottom of the file to see why type_id 3800 is missing. Again, if you can't solve it, replace the object and try again.

Don't overlook the error log. It can be helpful in pointing you in the right direction anytime CFS2 crashes or a mission won't load and run properly.

Summary

Unless you are using a purely stock installation, you will encounter errors from time to time along with the occasional CTD. While they can be frustrating, don't give up. Consider what you were doing last and try to retrace your steps to locate the problem. Hand-editing, for all its advantages, can also be the source of lots of problems, so take it step by step and be methodical in your approach. Recently more and more of us have been experimenting with adding additional infrastructure to the GSL and this is opening a new can of worms with things like error messages related to memory. All the more reason to tread lightly and back up frequently.

In the end, I recommend that you not spend too much time trying to solve a corrupted mission problem; if the answer doesn't come easily, scrap the mission and write a new one. After all, that's the great advantage of the Mission Builder, you can always write another one.

Part four

Campaign building and beyond

A step-by-step guide to building a campaign in CFS2 and a look at some of the other related files associated with mission building

Chapter seventeen – writing an effective campaign

Preflight

By now you are probably feeling pretty proficient as a mission builder and you're looking for a way to string all of those great missions together into a campaign. In this chapter I will take you step by step through the campaign building process. When you're done, you'll be able to build your own campaigns, using your own missions. Whether simple or complex, long or short, you'll find everything you need to get you started in this chapter.

Throughout this chapter I will be using several terms that may be new to you. They are defined here to help in your understanding of the process.

Branching A type of node that sends the player off in a different direction. A branching node will send the player to one of two other nodes based on random chance or the player's skill. A campaign using branching nodes could play differently every time it is flown depending on the number and complexity of the branching nodes used. See Dynamic.

Condition Something that is essential to the occurrence of something else. In campaigns, the completion of a mission causes a condition to occur that will either be true or false. That condition will determine the next mission to be flown.

Dynamic A type of campaign that uses branching to move the player around in different ways each time it is flown. A dynamic campaign could use a combination of random chance, player skill, and static nodes to produce an endless variety of options for the player. The stock campaigns are dynamic campaigns. See Branching.

Linear A campaign that travels in a straight line from the first mission to the last mission. In a linear campaign the player always starts at mission number one and progresses in order to the last mission every time the campaign is played with no variation. See Static.

Node The essential element of a campaign that determines which mission is to be flown and where the player will go next upon completing the mission. A campaign is a collection of nodes. Nodes can be either branching or static.

Static A node that is not branched and that always sends the player to one other specific node. A campaign comprised only of static nodes would be a linear campaign. See linear.

First we'll look at building a simple linear campaign. Following that, we'll move on to a dynamic, branching, campaign. Finally, I'll show you how to add custom movies and sounds to your campaign and give you some tips on troubleshooting your campaign.

The Linear Campaign
Campaigns are nothing more than a string of missions all hooked together. Any mission that you can fly in CFS2 can be used in a campaign. There is no structural difference between a single mission and a campaign mission. The difference between the two is location. Single missions are stored in the **MISSIONS** folder within CFS2. Campaign missions are stored in the **CAMPAIGNS** folder. For your campaign to work, all of the missions it uses must be stored in the CAMPAIGNS folder.

If you have a batch of missions that you like to fly as single missions but you also want to make a campaign out of them, simply COPY (not move) the missions to the CAMPAIGN folder. That will leave a set in both folders and you can then fly them as either single missions or as a campaign. The same is true in reverse. If you download a campaign and you'd really like to fly some of the missions as single missions, simply COPY (not move) the missions you're interested in into the MISSIONS folder and they will show up on the Single Missions menu.

The heart of a campaign is the campaign file. This is a text file that links all of the missions together. Campaign files must be stored in the CAMPAIGNS folder and they have the extension **CMG** after the name. The two stock campaign files are named ***war in the pacific – ijn.cmg*** *and* ***war in the pacific – usn.cmg***. Locate them in your CAMPAIGNS folder. We will look at them in detail later. The campaign file is a simple text file and can be read or created using Notepad.

The Campaign Template
Included with this Handbook was a sample campaign file named **CC_HB_TEMPLATE**. Open that file now using Notepad. This is a very basic and simple template for building a linear campaign. You can store this template in your CAMPAIGNS folder, or anywhere else that is easy for you to find, and use it over and over again as the starting point for your campaign building. **NOTE:** If you would prefer not to use the template you can simply open Notepad and start with a clean screen and type in all of the information from scratch. Once opened, the **CC_HB_ TEMPLATE.cmg** file looks like this:

```
[Campaign Global Info]
version=2.0
campaign_name=%campaign_name_string%
overview=%campaign_overview%
allegiance=(Enter nationality number here)

[node.0]
true_branch=1
percent_true=100
filler=0

[node.1]
mission_name=( enter first mission file name here)
true_branch=2
percent_true=100
filler=0

[node.2]
mission_name=( enter second mission file name here)
true_branch=3
percent_true=100
filler=1
```

[node.3]
mission_name=(enter third mission file name here)
true_branch=4
percent_true=100
filler=1

[node.4]
mission_name=(enter last mission file name here)
percent_true=100
filler=1

[Strings]
"campaign_name_string"=(Enter the title of the campaign here)
"campaign_overview"=(Here is where you enter an overview of the campaign. This is what the player will first see when selecting the campaign. Give a brief synopsis of what the campaign is about and watch your spelling. Remember, there is no spell check in Notepad.)

The campaign file is broken down into three sections, the header, the nodes, and the strings section. Let's look at those three sections.

The header section contains some basic, mandatory information that your campaign must contain in order to run.

[Campaign Global Info]	Must have this line to run
version=2.0	Must have this line to run
campaign_name=%campaign_name_string%	Points to the campaign name entered below
overview=%campaign_overview%	Points to the campaign overview entered below
allegiance=(Enter nationality number here)	Indicates the nationality under which this campaign will show up.

The node section contains the individual missions to be flown and indicates where you will go next. The first node, [node.0], does not have a mission attached and points either to a single node (usually node.1) or acts as a splitter node.

[node.1]	The node number. Node numbers should be in sequence starting with [Node.0]
mission_name=(enter first mission file name here)	The file name of the mission, not the title of the mission that appears in the briefing.
True_branch=2	The node, 2, that the player will advance to when the condition is true.
Percent_true=100	The probability that the condition will be true expressed as a percent. In this case it will always be true (100%).
Filler=0	Whether or not the player can enter the campaign at this point. A "1" means no and a "0" means yes. This MUST be set to "0" for [node.0] or you won't be able to start the campaign.

The Strings section contains the name of the campaign as it will appear in the game and the overview. The overview is the first thing the player sees and should give a brief synopsis of what the campaign is about.

[Strings]
"campaign_name_string"=(Enter the title of the campaign here)
"campaign_overview"=(Here is where you enter an overview of the campaign. This is what the player will first see when selecting the campaign. Give a brief synopsis of what the campaign is about and watch your spelling. Remember, there is no spell check in Notepad.)

In the above example I have highlighted the text that you will change in red to make it easier to see. Now that you have seen the basic campaign structure, let's fill in the details.

Creating Your Campaign
First, enter next to "**allegiance=**" the country number for the nationality that you want this campaign to fly for. The number you enter here coincides with the country numbers in your **country.cfg** file found in the main CFS2 directory. USA is number 3, Japan is 4, Britain is 1, Germany 2, Australia 10, etc. This number has nothing to do with the nationality that you selected for the pilot when you wrote the mission, only the country the campaign will show up under when you choose it in the game. You could write a campaign flying Zeros for Japan but if you enter a "2" in this spot you'll find the campaign listed under Germany, not Japan.

I will design a fictional campaign flying for the Black Sheep Squadron. There will be only four missions in my campaign. The file names for the missions are blk_sheepA through blk_sheepD. Since I am flying for the USA I will enter a "**3**" as my allegiance: **allegiance=3**

Next you will enter one node for each mission in the campaign. In my example I have four missions so I have five nodes, the first node [node.0] has no mission attached to it. Your campaign may have more or fewer missions. Add or delete nodes as necessary. **NOTE:** The first node in the campaign [node.0] does not have a mission attached. It contains only the node number, true_branch, percent, and the "filler=0" line. Node.0 is used to point the player to the node where you want them to start flying. It can point to a single node or be used as a splitter node. **NOTE:** The last node in the campaign does not have a true_branch. It contains only the node number, mission name, percent, and the "filler=1" line. For your campaign to end properly you MUST follow this format. If you leave a "true_branch=" in the last node the campaign file will be corrupt and will not load.

Finally, add a name for your campaign and an overview of what the campaign is about. Here's my sample. Yours will look somewhat different:

[Campaign Global Info]
version=2.0
campaign_name=%campaign_name_string%
overview=%campaign_overview%
allegiance=3

[node.0]
true_branch=1
percent_true=100
filler=0

[node.1]
mission_name=blk_sheepA
true_branch=2
percent_true=100
filler=0

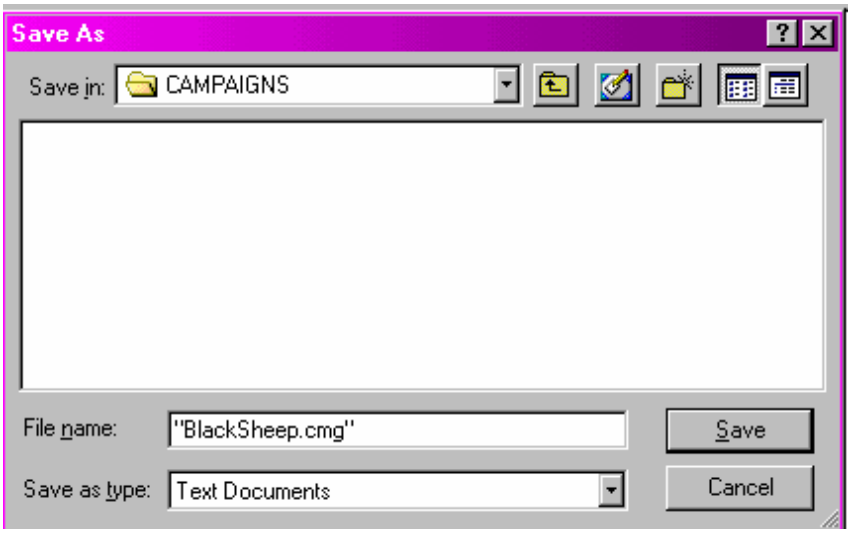
[node.2]
mission_name=blk_sheepB
true_branch=3
percent_true=100
filler=1

[node.3]
mission_name=blk_sheepC
true_branch=4
percent_true=100
filler=1

[node.4]
mission_name=blk_sheepD
percent_true=100
filler=1

[Strings]
"campaign_name_string"=Black Sheep Squadron
"campaign_overview"=Fly with "Pappy" Boyington and his Black Sheep Squadron as they challenge the Japanese in the skies over the South Pacific.

Since this is a simple linear campaign it is comprised of static nodes. Each node points to the next node in sequence and the condition is always true. In other words, mission #1 will always be flown first, 100% of the time you will go to mission #2, followed by mission #3, followed by mission #4. Look at the nodes and you'll see what I mean. This campaign will always follow the same pattern.



To save your campaign click **File – Save As…** In the **Save in** window go to the CAMPAIGNS folder in CFS2. Notice that in the drop-down **Save as type** window there is no **cmg** file format. In order to save this file as a campaign file you must follow these steps: Enter the name of the campaign file followed by a period and the extension **cmg** in the *File name* window and enclose the entire name and extension in quotation marks. This will now save your file as a campaign file. If you fail to add the **cmg** extension and enclose the whole thing in quotation marks it will save the file as a text file and it will not work in the game. See the illustration at the left.

That's it! You have just created your first campaign. Easy wasn't it? Once you've done one, it becomes very easy to construct linear campaigns. Just be sure to double check your numbering so that each node leads properly to the next one in sequence.

Linear campaigns are fun and easy but after flying the same set of missions over and over you'll probably get bored. To break that monotony, build a dynamic campaign.

The Dynamic (Branched) Campaign
If you got this far and you are a little confused or you had trouble building your first linear campaign, stop. Go back and review the previous section. You need to understand the basic linear campaign before jumping into branching. If you sailed through the linear campaign, then read on.

The only difference between a static/linear campaign and a dynamic/branching campaign is the nodes. Everything else is the same; the header, strings, and basic format do not change. Only the design of the nodes is different.

True and False
After completing a mission the condition can be either True or False. In a static node the condition is always true and the percent is always 100. In plain English that means that:

"If the condition is true send the player to the next node. The condition will be true 100% of the time".

In a branched node the condition can be either True or False. If the condition is true the player advances to node *"x".* If the condition is false the player advances to node *"y"*. There are two ways to decide whether the condition is true or false, random chance or the player's skill.

Random Chance
Perhaps in your campaign you come to a point where you want the player to fly one of two different missions. Maybe both missions are similar and to fly them both would be boring or maybe you just want to add some variety to the pattern of the campaign. Either way you don't care which mission is flown just so long as the player files only one of them; to achieve that we will use a branched node and a random percent. The node might look something like this:

```
[node.25]
mission_name=blk_sheep_aa
true_branch=26
false_branch=27
percent_true=50
filler=1
```

Notice that the branched node has an additional line, a ***false_branch=*** line. Also notice that we have reduced the percent true from 100 to 50. What this node now says is:

"If the condition is true send the player to node 26. If the condition is false send the player to node 27. The condition will be true 50% of the time."

In plain English, half the time the player goes to node 26 and half the time he goes to node 27, and purely random chance will determine that. That was easy! You can make the percent true any number between 1 and 100. If you set it at *percent_true=75*, then 75% of the time, or three out of

four times, the player would go to node 26 and 25%, or one fourth of the time, the player would go to node 27. By adjusting the percentage above or below 50 you will favor one mission or the other in the random selection. When put together this section of the campaign file will look something like this:

```
[node.25]
mission_name=blk_sheep_aa
true_branch=26
False_branch=27
percent_true=50
filler=1
```

The player completes the mission and goes to either node 26 or 27.

```
[node.26]
Mission_name=blk_sheep_ab
True_branch=28
percent_true=100
filler=1
```

The player has a 50% chance of landing here. When completed the player goes to node 28.

```
[node.27]
Mission_name=blk_sheep_ac
True_branch=28
percent_true=100
filler=1
```

The player has a 50% chance of landing here. When completed the player goes to node 28.

```
[node.28]
mission_name=blk_sheep_ad
True_branch=29
percent_true=100
filler=1
```

Regardless of which node the player flew he ends up at this point.

Player Skill

The other way to determine which node the player advances to is through the skill of the player. Obviously this has some advantages over random chance depending on the complexity of your campaign and what it is that you are trying to accomplish. Player skill is determined by the number of goals the player completes in the mission.

You'll recall that you can set goals within a mission. Every mission has one automatic goal – the player must survive. But beyond that you can set goals for the number of planes shot down or ground targets destroyed, etc. You can mark specific infrastructure or ships as having to be destroyed, making them a goal or you can make an event a goal by marking it as "Count as Mission Goal". The Mission Info panel in Mission Builder will tell you the total number of goals your mission has. To use goals in branching we replace the ***percent_true=*** line with a ***goals=*** line. The result might look like this:

```
[node.25]
mission_name=blk_sheep_aa
true_branch=26
false_branch=27
goals=4
filler=1
```

Notice that the only change we made was replacing ***percent_true=50*** with ***goals=4***. Now this node says:

"If the condition is true send the player to node 26. If the condition is false send the player to node 27. The condition will be true if the player completes at least four goals."

"The condition will be true if the player completes at least 4 goals." It doesn't say that the player has to complete **ALL** of the goals, only that he must complete at least 4 of them. If your mission only has 4 goals (remember one of them is that the player has to survive) then he would have to complete all of them to advance to node 26. However, you might have a mission with 8 goals and you'd be satisfied if the player completed only 4 of them. In that case, should the player complete 4 <u>or more</u> of the goals he moves to node 26. If he completes less than 4 goals he goes to node 27.

If you use goals to branch a node be sure that you always have a minimum of 2 (goals=2). Using the line *goals=1* is useless because the one goal is that the player must survive and if he doesn't survive the campaign ends right there. The completed section of the campaign file using goals might look like this:

```
[node.25]
mission_name=blk_sheep_aa
true_branch=26
False_branch=27
goals=4
filler=1
```

The player completes the mission and goes to either node 26 or 27.

```
[node.26]
Mission_name=blk_sheep_ab
True_branch=28
percent_true=100
filler=1
```

If the player completes 4 or more goals he lands here. When completed the player goes to node 28.

```
[node.27]
Mission_name=blk_sheep_ac
True_branch=28
percent_true=100
filler=1
```

If the player doesn't complete 4 goals he lands here. When completed the player goes to node 28.

```
[node.28]
mission_name=blk_sheep_ad
True_branch=29
percent_true=100
filler=1
```

Regardless of which node the player flew he ends up at this point.

Mix and Match

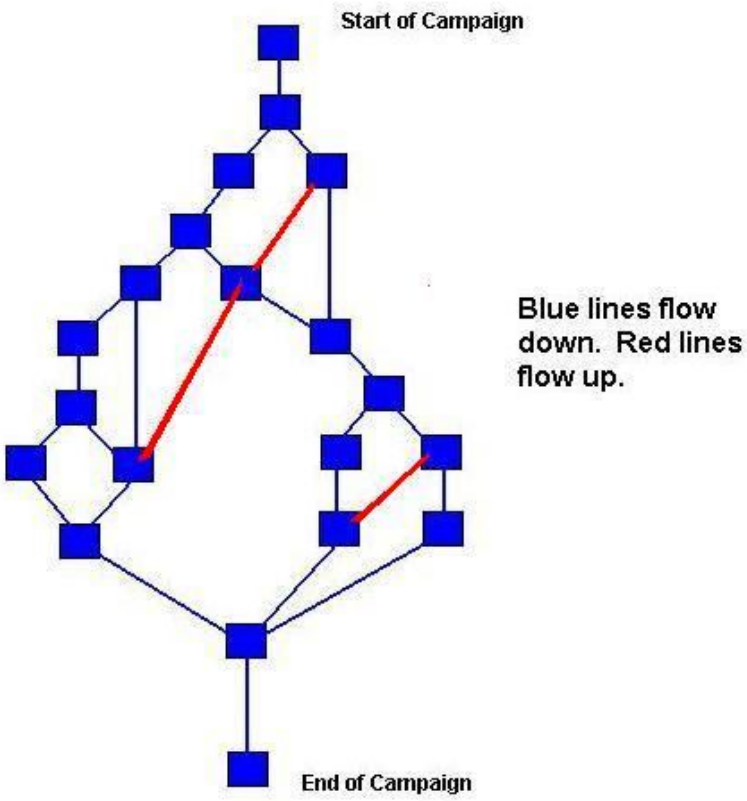
You can't use both random percent and goals in the same node but you certainly can use them both in the same campaign. You'll probably find that the best campaigns are a combination of static nodes and branching nodes that use both random chance and goals to determine the condition. Here's a sample of a section from an actual campaign file that combines all three.

```
[node.2]
mission_name=SAAF_02
true_branch=3
false_branch=4
goals=2
filler=1

[node.3]
mission_name=SAAF_03
true_branch=5
percent_true=100
filler=1

[node.4]
mission_name=SAAF_04
true_branch=5
percent_true=100
filler=1

[node.5]
mission_name=SAAF_05
true_branch=6
false_branch=7
percent_true=50
filler=1

[node.6]
mission_name=SAAF_06
true_branch=9
false_branch=8
goals=2
filler=1

[node.7]
mission_name=SAAF_07
true_branch=9
false_branch=8
percent_true=50
filler=1
```

Notice that in this short six-node segment of an actual campaign there are 2 static nodes and 4 branching nodes, two each of goals and random percent. Even when this campaign is played repeatedly it offers significant variety to the player.

Using your imagination you can see how a campaign can very quickly branch over and over again and give you an almost infinite number of options depending on the size of your campaign. The pitfall with all of this variety is that you can get lost very easily. If you have too many branching nodes you run the risk of getting into a loop or creating a dead end, either of which will ruin you campaign. I would encourage you to map out the branching movements on a piece of scratch paper so that you can keep track of which nodes link to which and how you want them to play out.

I also create a campaign summary for every campaign I write. I fill it in as I finish writing each mission and update the campaign file. It allows me to keep track of the node and mission number, the date of the mission (so I can keep things in chronological order) and a brief synopsis of the mission. This is not a briefing that the player sees so my synopsis covers what actually happens in the mission and what I am trying to accomplish with my goals. I also note where the player is supposed to go next. When things get goofed up it allows me to quickly see on one page what is supposed to happen without having to open a lot of individual mission files.



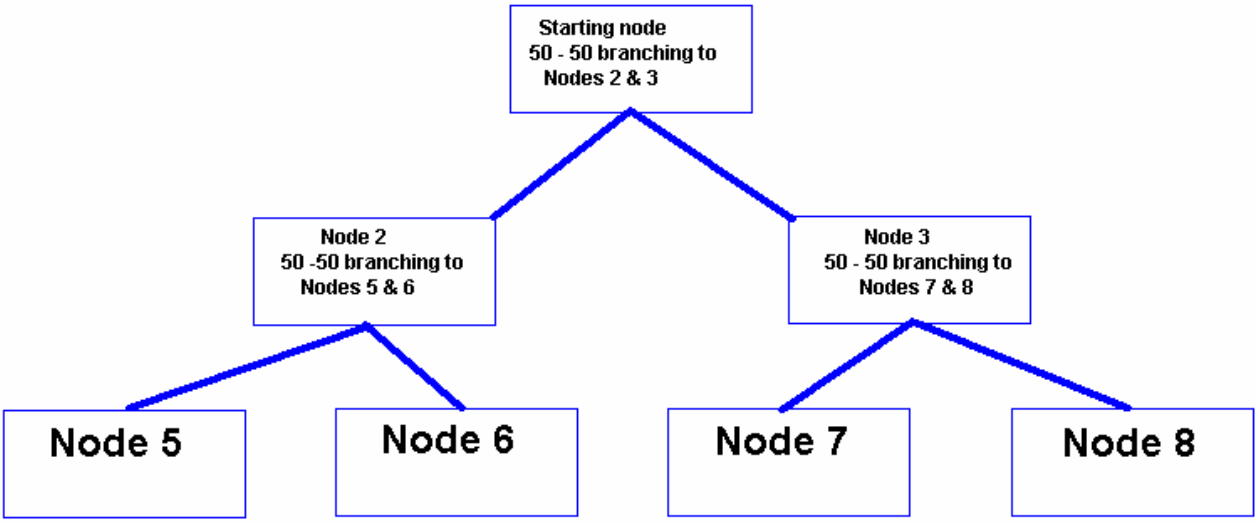
Remember too that no matter how many branches you have in your campaign you can always bring everything back together by directing all of the branching nodes back to one central point, such as the final mission.

Here is a chart of a 20-mission campaign. Notice the variety that is built in using both branching and static nodes. Also note the dual direction of the flow. Blue lines flow down from the start of the campaign to the end. Red lines represent nodes that point back up toward earlier nodes in the campaign. Notice too that seeing the campaign in chart form helps you keep track of connecting the nodes. Depending on how it plays out, this campaign could be as short as 9 missions or as long as 18.

Multiple Branching

As you saw earlier, a branching node can only branch in two directions. Sometimes you may want to branch off in more directions then just two from a single mission. That's easily accomplished by using multiple branching. In multiple branching, the node you originally branch to branches again, using a **Splitter Node**, before running a mission. A Splitter Node does not have a mission attached. It merely performs a branching function.

Say the player reached a point in the campaign where you wanted him to jump to any one of four different missions. To accomplish this you will use three nodes to create the four-way branch. Look first at the illustration to see what we are planning to do:



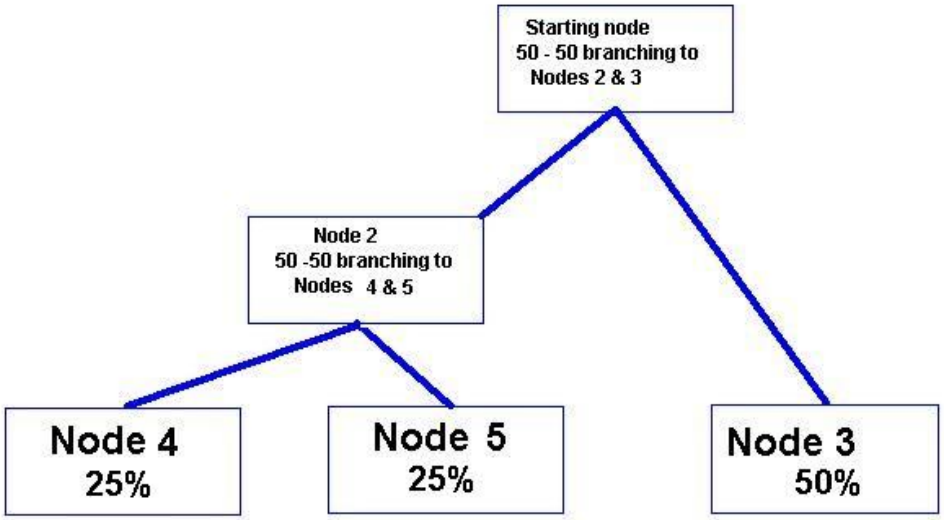
Here's how we write it up in the campaign file:

[node.1]	
mission_name=SAAF_01	After completing this mission the branching starts
true_branch=2	Node 2 above – Splitter Node
false_branch=3	Node 3 above – Splitter Node
percent_true=50	
filler=1	
[node.2]	Splitter Node – note there is no *mission_name* line
true_branch=5	Node 5 above – one of 4 possible missions
false_branch=6	Node 6 above – one of 4 possible missions
percent_true=50	
filler=1	
[node.3]	Splitter Node – note there is no *mission_name* line
true_branch=7	Node 7 above – one of 4 possible missions
false_branch=8	Node 8 above – one of 4 possible missions
percent_true=50	
filler=1	

When the player completes the mission at [node.1] he has a 50-50 chance of going to either [node.2] or [node.3]. As soon as he gets to one of those nodes he branches again, to either [node.4], [node.5], [node.6], or [node.7]. Of course all of this happens in the blink of an eye. From the player's perspective he completes one mission and has an equal, but random chance of going immediately to any one of four other missions.

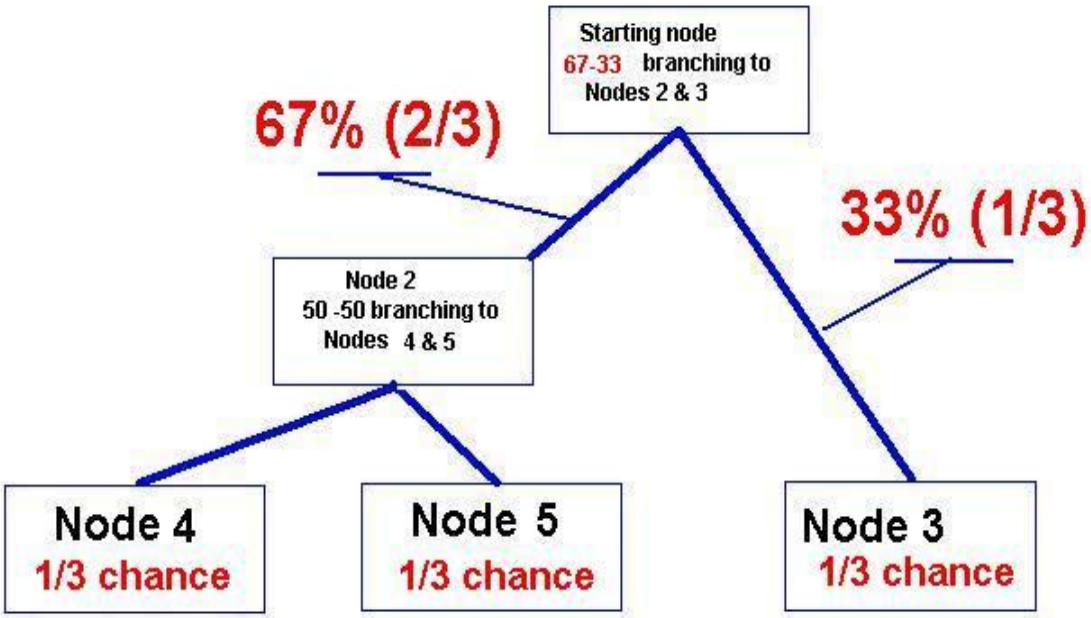
You can see how using Splitter Nodes can quickly branch your campaign off into numerous directions. You can also probably see how easy it is to get confused if you don't plan ahead and think things through carefully.

It's easy to do multiples of 2 since we always have two choices, true or false. But what if you want to do a 3-way split? A three way split is constructed just like a four way except you use only one splitter node instead of two.



The problem is that this won't create a balanced split if you use the standard 50-50 branching. Half the time the player goes to Node 3 and the other half of the time he goes to <u>either</u> Node 4 <u>or</u> Node 5. This weighs the split in favor of Node 3.

The solution is some simple high school math. Instead of splitting 50-50, do a two thirds/one third split at the Starting Node then a 50-50 split at Node 2. Charted out it will look like this:



This gives the player an equal chance of advancing to any one of three different missions. Due to rounding there is a slight, statistically insignificant, advantage for Node 2.

Written into the campaign file the balanced split looks like this:

[node.1]	
mission_name=SAAF_01	
true_branch=2	Node 2 above – Splitter Node
false_branch=3	Node 3 above – one of 3 possible missions
percent_true=67	
filler=1	
[node.2]	Splitter Node – No mission attached
true_branch=4	Node 4 above – one of 3 possible missions
false_branch=5	Node 5 above – one of 3 possible missions
percent_true=50	
filler=1	

By experimenting with different percentage values and using Splitter Nodes you can add infinite variety to how your campaigns play out.

Movies and Sounds

Go to your CAMPAIGNS folder in your main CFS2 directory and find the campaign file ***war in the pacific – usn.cmg*.** Open it using Notepad. This is the stock U.S. Navy campaign that comes with the game. It is a dynamic campaign using many branching nodes. Take a few moments to look at the campaign structure but DON'T CHANGE ANYTHING!

The first thing that you'll notice is that there is some extra information between the allegiance data and the first node that we didn't include in our campaign. Ignore this data for now. Notice that [node.0] is a Splitter Node. As soon as you start the game you have an equal chance of going to either [Node 1] or [Node 4]. If you'll look at those nodes you'll see that they are Splitter Nodes as well. Microsoft has done what we did above and created a balanced four-way split right at the start of the campaign. This gives the player four different starting points into the campaign.

This campaign is fairly large, 154 nodes. As you scroll through, notice that they use a combination of static and branching nodes, random chance and goals. You'll also notice that a number of nodes have sound and movie files associated with them. These files are referenced by the lines **soundfile=** and **moviefile=**. You see and hear these sound and movie clips at the start of the campaign, at certain missions along the way, and at the end if you successfully complete the campaign. Scroll back to the beginning of the campaign file and focus in on [node.2].

```
[node.2]
mission_name=hn1a_usn
true_branch=7
percent_true=100
soundfile=pt001.wav
moviefile=ptadgj.mpg
filler=1
```

Node 2 is one of the starting points for the campaign and has both a sound and movie file associated with it. When you first begin the campaign the game will play a movie and you'll hear a voice-over explaining the war in the Pacific. The command lines *soundfile*= and *moviefile*= tell the game which movies and sound files to play. All movie files used by the game are in mpg format and are stored in the folder **CFS2\videos\transit**. The sound clips associated with these movies are wav files and are stored in the folder **CFS2\SOUND\VO**. The sound files also have a text file associated with them that scrolls the voice-over text below the movie so that you can read along during the narration.

Sound and movie files can be added, modified, or deleted from your campaign. You don't need either to write an exciting and workable campaign. However, if you decide to use sound and movie files in your campaign there are a couple of things to know. You **must** have a movie file if you want to have a sound file. You can have a movie file without a sound file and the movie will play just fine, but you cannot have a sound file without a movie file. The game will just skip the sound file entirely and move on as though it wasn't there.

You can use your own sound and movie files as long as they are created in the **mpg** and **wav** formats and they are stored in the proper directories. You can also mix the stock movies with your own sounds or your own movies with the stock voice-overs. The stock movies have background music associated with them that plays as a part of the movie. You can achieve some interesting effects by showing the stock movie with your own sound effects (air raid sirens, bomb blasts, even some of the stock radio transmissions) as the voice-over instead of using the stock sound files. Those of you that are especially talented could create you own mpg movies or record your own voice-overs to add into your campaign.

Looking again at the stock campaign, note the two sections directly above [node.0].

```
[failure]
moviefile=
soundfile=

[success]
soundfile=pt011.wav
moviefile=ptk.mpg
```

These two sections are used to call up movies and sounds when you either successfully complete or fail the campaign. The two are self-explanatory. Notice that the game shows you a movie when you succeed but not when you fail, although you could add one to the stock campaign or use this format in your own campaigns.

Troubleshooting your Campaign

No matter how hard you try, in a complex campaign something may go wrong. Unfortunately you might not discover the problem until you are well into the campaign and it suddenly crashes. Here are a few of the problems that I've discovered and how to avoid them.

<u>Size matters!</u> If you are running an older operating system, such as Windows 98se, the maximum size for a **cmg** file is 65kb. That is a huge amount. To create a campaign file that large the campaign would have to have around 700 nodes and you'd have to use WordPad or some other large text editor because that's too big for Notepad. I can't imagine anyone writing a campaign that big so this will probably never be an issue. If you are running a newer operating system, such as XP, the size limit won't apply.

Can't select a pilot The campaign name appears in the campaign selection window but you can't create a pilot and start the campaign. This is caused when you have a mission assigned to the first node, [node.0]. Node 0 contains only the node number, true_branch, percent, and the "filler=0" line. Node.0 is used to point the player to the first mission you want them to fly. It can point to a single node (usually [node.1]) or be used as a splitter node.

Campaign won't load If, after you create your pilot, the campaign won't load and you get an error message on the screen, the file is corrupt. There are several causes. It could be that there is no ending to the campaign. The last node in the campaign must not have a true or false branch. It should consist of only three lines, the mission, percent_true=100, and filler=1. Or it could be that there is no beginning.

The first node, [node.0], must have the line *filler=0* so that you can start the campaign. Also, the *percent_true* must equal 100. If it is less than 100 and it comes up false your pilot will show as "retired" as soon as he is selected and the campaign won't load. Oddly enough, if you leave the *percent_true* line out altogether in [node.0] the campaign treats that as a true condition and loads normally.

Another cause for this is reference to a non-existing node. If you have a node that points to [node.34], and there is no [node.34] in your campaign it will cause this error. Also, having two nodes with the same number will cause the campaign not to load. In other words if you had two [node.10]'s.

Campaign stops running You're flying along completing mission after mission and all of a sudden after another successful flight you find yourself "retired" and the campaign over. A couple of things could cause this. If you have a node that doesn't equal 100% it can end your career. Maybe you made a typo in a static node. You meant to type **percent_true=100** but you actually typed **percent_true=10**. When you complete that mission there is now only a 10% chance that the condition will be "True" and that you will move on. If the condition is "False" and there is no false statement in the node the campaign ends and you are retired.

Other typos within the campaign file will also cause this; for example, leaving out an underscore. Typing **true branch=4** instead of **true_branch=4** will get you an early retirement.

Long campaign quits before the end If you write an exceptionally long campaign it is possible that you are flying along, completing mission after mission, and all of a sudden find your self "retired" and the campaign terminated. This can happen when you have lost too many wingmen, you have lost too many player aircraft, or your mission success percentage is too low. There are two ways to prevent this. First – be a better pilot! If that isn't working you can hand-edit your pilot's log.

Go to the folder **PILOTLOG** in your main CFS2 directory and look for the files associated with your pilot. You'll see two files with your pilot's name. One is a text file, ignore it; the other is a **0** file, open it using Notepad. This file contains all of the critical information regarding your performance in the campaign. If you look closely you will see that it contains your rank, medal count, campaign score, number and type of kills, etc. as seen here:

```
[pilot info]
side=0
allegiance=10
difficulty=0
```

[country.1]
rank=4
num_medals=5
score_total=71331
purple_heart=1
fighters_total=106
bombers_total=176
ground_total=312
voicepack=0
skill=0

[campaign.0]
campaign_name=SAAF
next_mission=SAAF_JB_479
next_node=509
missions_completed=124
lowscore=0
wingmen_lost=5
player_aircraft_lost=1
current_score=98

You can edit any of these fields but the three that can get you dumped out of the campaign are shown in bold. If you want to prevent getting bounced, open this file and reduce these numbers periodically. The section shown above is followed by a synopsis of your activity in every mission you've flown. At the end of that is a section for your wingmen. You'll see entries something like this:

[wingman.0]
name=Trevor Cook
fighters_total=7
bombers_total=2
ground_total=26
voicepack=12
skill=2

The key field here is the skill level. You can edit this to raise or lower the skill level of any particular wingman. So, if you're tired of them flying around in circles instead of fighting, you can modify all of your wingmen to be aces (**skill=3**).

Other errors You wrote this great campaign and now you can't find it anywhere. Check the allegiance line at the beginning of the campaign. You may be looking for the campaign under the wrong nationality or you may have typed in a number that doesn't coincide with any nationalities in your country.cfg file.

If the mission referenced in the node doesn't exist or the game can't find it you will get a reject message and the campaign will quit. Maybe you typed in the mission file name wrong or maybe you put the mission in the MISSIONS folder instead of the CAMPAIGNS folder.

You find yourself repeating the same mission over and over again. Either you've stumbled across a rift in the space-time continuum or you've created a loop in your campaign. This is another good reason to document or chart out your nodes on paper. It's easy to send the player back to an earlier

node but doing so could get you into a never-ending loop of flying the same missions over and over. Follow the logic of each node to detect the error.

Loop backs are not necessarily bad. Maybe you have a mission where the goal is to sink a battleship. If the player completes the goal you want them to advance. If he doesn't complete the goal you want him to try again. Using a branching node and goals you can construct this situation. It might look like this:

[node.15]	
mission_name=SAAF_21	The mission to sink the battleship
true_branch=16	If the ship is sunk the player moves on
false_branch=15	If the player fails they repeat this mission
goals=2	Goals: 1)survive 2)sink the battleship
filler=1	
[node.16]	
mission_name=SAAF_22	The next mission after the battleship is sunk
true_branch=17	
percent_true=100	
filler=1	

A variation on this would be to have the player go to a third node if he failed to sink the battleship. Instead of repeating the same mission the third node might start out with the battleship damaged. It could more accurately simulate failing to sink the ship, rearming, and going back for a second try.

Summary

Building your own campaign can add a lot of additional fun to what already is a great game. Your campaign can be ultra historically accurate, totally fictional but realistic, or not make any sense at all. The key is to create a campaign that YOU enjoy and that is fun for you to fly.

Maybe you have a dozen missions that you just really like so you decide to string them together in a campaign. They could be flying for different nations, in different parts of the world. It doesn't matter. Another idea would be to create a linear campaign where you fly the mission the first time as a U.S. pilot then fly the same mission from the other side as a Japanese pilot. Using Mission Builder you could construct mirror images of each mission for the two nationalities.

Granted, in campaigns such as these the awards and promotions wouldn't make any sense but that doesn't matter if the campaign is fun for you to fly. Speaking of awards and promotions, there is nothing in the campaign file that controls what awards you receive or when you get promoted. That is handled elsewhere within CFS2 and is outside the scope of this chapter.

As you've seen, building a campaign can be easy or complex. The larger your campaign gets and the more branching nodes you use the greater the chance for error. I strongly encourage you to document what you are attempting to do so that you can visualize the flow of events. Pay attention to detail if you want to succeed.

Just like mission building, when you are writing a **cmg** file you are essentially writing computer code. Computers are dumb. They only understand one language – their own. When you talk to a computer you must learn to talk to it in its own language. You must talk to it using only the words it will understand. If you merely get close, the computer isn't smart enough to guess what you are trying to

say. Accuracy counts. When using Mission Builder you are using a graphical interface that converts plain language and symbols into the code for you. With the **cmg** file you must be much more precise. A misplaced quotation mark or left out underscore can result in no campaign. Remember the lessons from our chapters on hand-editing. Work slowly, carefully and double check your work.

Campaign building is sort of the apex of mission building. After creating all of those great missions you can now string them together in an exciting and ever changing campaign that will give you many hours of pleasure.

Customizing Your Wingmen Names

Ever wonder where those wingmen names come from in a campaign? They originate in the **country.cfg** file found in your main CFS2 folder. Find this file and open it with Notepad. Scroll down toward the end of the USA entry and you'll see a section called **first_names=,** followed by a bunch of first names and a section called **last_names=**, followed by a bunch of last names. When CFS2 generates a new wingman for you it randomly grabs a first and last name and combines them to create your new pilot's name. That's OK most of the time, but what if you want to use names of real people as your wingmen?

Entering real names into the **country.cfg** file isn't a good idea because the game will still grab one name from each section. So, if you entered Butch O'Hare as a first name it might come out as Butch O'Hare Jones in your campaign. The trick is to hand-edit your **pilotlog** as shown in the troubleshooting section above. Remember our wingmen section:

[wingman.0]
name=**Trevor Cook**
fighters_total=7
bombers_total=2
ground_total=26
voicepack=12
skill=2

If your don't want Trevor Cook as your wingman simply edit this name to anything you want and that person will remain your wingman throughout the campaign, unless they get shot down.

I don't know about you, but I always feel a lot better having Joe Foss and Pappy Boyington flying on my wing!
☺

Chapter eighteen – other related files

Preflight
No discussion of mission building would be complete without a closer look at some of the other CFS2 files that interact with the Mission Builder. In some cases these files are used exclusively by Mission Builder, in other cases they affect the game as well. All of them can be hand-edited by you to improve your CFS2 experience. In this chapter we'll look at some of those files, how they interact with the Mission Builder, and how to edit them to enhance your gaming experience.

airbases.dat
The airbases.dat file is located in the **INFO** folder within CFS2 and contains a list of the runways that can be used in Mission Builder. Only runways that have an airbases.dat entry will show as an "X" on the Mission Builder map and can be used for the **Take Off** or **Land at Base** commands.

When you download and install a new airbase into CFS2 the files that make up that scenery contain the location of the runway, visual scenery, and your starting point for taking off in Free Flight. Once properly installed and activated the airbase will be available for use in Free Flight. It won't be available for use in Mission Builder until you add it to the airbases.dat file. The airbases.dat file is pretty simple to figure out and is easily modified. Open it with Notepad and take a look.

```
[runway.0]
id=9000
name=*Abbeville
heading=322
allegiance=0
units_across=2
base_lat=N50 9.97
base_lon=E1 46.44
base_alt=118
takeoff_start_pos=-600,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=600,0
landing_stop_pos=-250,0
```

The information shown in **bold** is the information you will modify for each new entry.

The first line is the runway number. The numbers must start with **[runway.0]** and progress sequentially through your last entry. If you break the sequence of numbers, all runways below the break will not show up in Mission Builder.

The second line is the runway **id** number. As we discussed earlier in this Handbook, there is no master list for all runways worldwide. Once you go past the stock entries you are free to make up your own id numbers. Some downloaded airbases come with airbases.dat entries and id numbers assigned by the designer. You can use any four digit number just remember one rule: id numbers should NEVER be repeated. Duplicate id numbers can cause CTD's and other problems. Refer to Chapter Fifteen – *How to Take Off from any Point on the Planet*, for more information on id number problems.

The next line is the runway name as it will appear in Mission Builder. Since you are entering this name it may or may not match the name shown in Free Flight; that doesn't matter. Note the asterisk (*****). A special character such as this will appear before an alpha character when sorting alphabetically. By placing an asterisk in front of all of the CFS1 European runway names Microsoft has insured that they will appear first on the drop-down menu.

The heading is self explanatory and will be discussed in more detail below.

This allegiance number will match those entered in your **country.cfg** file. A zero, "**0**", as shown here is a neutral base, it can be used by any nation. If you enter a number for a specific nationality here then the runway will appear as either a Blue (friendly), or Red (enemy) "X" on the Mission Builder map depending on the nationality of the player. The problem with assigning a specific nationality is that only aircraft allied with that nation can be assigned to take off or land at that runway. To obtain maximum flexibility, leave the allegiance set at zero.

Units_across is the number of planes that can be lined up side by side at take off, well sort of. For 90% of the runways you enter leave this at **2**. That is the default and, as you can see in the game, works for almost every runway. If you make the units across too many planes wide those planes will start off the runway on the grass. Occasionally you need to reduce this to **1** for very narrow landing strips.

The exception here is if you have a WWI installation. In WWI, many of the runways were simply large grass fields and the planes took off in a line abreast. You can try to duplicate that by setting the units across to **8**. The end result is not perfect. Instead of a nice line of planes, your eight aircraft will start in an echelon left formation with the eighth in line some distance behind you. Still, it's the most realistic WWI take off you can get.

The base longitude, latitude and altitude are all self explanatory. As a general rule, the remainder of the entry can be copied from any of the stock airbases and used for your new airbase.

To enter a new runway you must obtain the exact location, runway heading, and altitude. To get that information, go into Free Flight and select the airfield you want to add. As you're sitting in the cockpit, before you start your engine, press **ALT**. Then select **Views**, **View Mode**, **Top-Down.** This will place you well above the airfield looking down. Your plane will change into a red cross. Press the "**Y"** key to enter slew mode. This brings up a line of text across the top of your screen showing the exact longitude, latitude, altitude and heading. Slowly push forward on your joystick to move the cross to the center of the runway. When you get to that point write down the longitude, latitude, altitude and heading as shown on the screen.

```
[runway.124]
id=8051
name= Hickam Field
base_lat=N21 20.26
base_lon=W157 57.65
base_alt=0
heading=050
allegiance=0
units_across=2
takeoff_start_pos=-400,0
takeoff_stop_pos=1600,200
landing_start_pos=5000,350
landing_touchdown_pos=500,0
landing_stop_pos=-250,0


▫
```

Once you have the data you need, open up the airbases.dat file. Scroll down to the bottom of the file and copy the last entry. Paste it in below the last line of text but above the little square box symbol. Now replace the information in the copy you just made with the new information you wrote down from Free Flight. Be sure to change the name of the airfield. You must also change the runway and **id** numbers. The runway number must be one more than the last entry. As noted above, the id number can be anything but it must be unique within the file. Always do a search of the airbases.dat file for the id number you are going to use to make sure you aren't entering a duplicate (Click Edit – Find and enter the number. Be sure to search both up and down). Make sure that the last thing below your new entry is a couple of blank lines followed by the little square. If you enter a runway below the little square it won't show up in Mission Builder.

In the example to the left I have added the runway at Hickam Field, Hawaii.

Some things to remember:

- Runway numbers **must** be sequential starting with **[runway.0]**. If you skip a number or otherwise break the sequence all runway entries below that point won't show up in Mission Builder.
- Runway id numbers should **never** be repeated.
- Do to differences between magnetic north and true north you may find yourself not lined up perfectly on the runway. If this is the case, simply obtain the runway heading from within Mission Builder by dragging the cursor and edit your airbases.dat entry to reflect the correct heading.
- The airbases.dat file must end with a couple of blank lines followed by the little square box.
- If your plane is sitting too far back, off the runway in the grass, you can adjust the **takeoff_start_pos=** line. In my example above the takeoff_start position is 450 yards short (**-450**) of the longitude and latitude settings that we entered. Since we are using the midway point of the runway as the centering mark you should never have a problem. If you are too far back, you can reduce this number to something like **–250**. Now you will start 250 yards short of center. If you want more runway to take off with, raise the number to something like **–800**.

That's really all you need to do to add a new runway to the airbases.dat file. If you want to, you can experiment with modifying the other numbers. Some of them have to do with controlling the AI on take offs and landings. However, as anyone who has experimented with trying to get the AI to land consistently knows, they pretty much do as they please without regard to anything else that is happening in the game.

☛ If you want to create a purely European installation of CFS2 you probably don't want all of those Pacific airbases showing up in Mission Builder and in your game. You can easily remove all reference to the stock Pacific bases.

To prevent them from showing up in Mission Builder simply remove all of the **airbases.dat** entries that reference them. In a stock installation that would be everything from **[runway.73]** thru the end of the file, **[runway.123]**. You can then reuse those runway numbers for any new European bases that you add.

To remove any reference to them in Free Flight and Quick Combat go to your **SCENEDB** folder and open the folder **afdfiles**. Inside you'll find another folder named **scenery**. Open it and delete the file **cfs2afd.bgl** located there. Finally go to the **runways** folder also located within **SCENEDB** and open the **scenery** folder found inside. Delete the file **allrunways.bgl**. You'll no longer see any hint of Henderson or any of the other stock Pacific locations.

CFS2\SCENEDB\afdfiles\scenery\cfs2afd.bgl – delete this file
CFS2\SCENEDB\runways\scenery\allrunways.bgl – delete this file

callsign.dat

The callsign.dat file is also located in the INFO folder and contains a list of the callsigns (names) that can be assigned to aircraft, ships and vehicles. We've already discussed how to add customized names for an individual mission or for your campaign wingmen. However, if you want more choices of names on the drop-down menu then you need to edit the callsign.dat file.

Open the file with Notepad. You'll see a simple explanation of the file at the top. Look at the American aircraft section and you'll see this:

```
# American aircraft
Agate,3,1
```

Agate is the callsign, **3** is the nationality (This number matches the nationality numbers in your country.cfg file), and **1** is the family number indicating an airplane. You'll note that right above the American entries is a list of the family numbers to use for planes, ships, and vehicles.

To add new callsigns just scroll down to the end of the list of American aircraft names and add your own. For example you might type in: **Cougar,3,1**. Now "Cougar" will show up as a callsign choice for American planes in Mission Builder. You can follow this same format for adding additional ship and vehicle names. Notice that ships include additional information. For example the Enterprise shows as:

```
Enterprise CV-6,3,5,13
```

Enterprise CV-6 is the callsign, **3** is the nationality, **5** is the family for all ships in this file and **13** is the category for aircraft carriers. Again, follow the existing format to add additional callsigns.

You can also add a whole new set of callsigns for any additional countries that you have added to the game. To do this, simply scroll down to the bottom of the file and follow the existing formats. Be sure to match up the nationality number correctly. Here is an example of some of my Canadian aircraft entries:

```
# Canadian aircraft
Beaver,16,1
Bison,16,1
Caribou,16,1
```

In my game Canada is nationality **16**, it may be something different in yours.

Another alternative is to download callsign lists that other players have created. There are several good ones floating around out there.

country.cfg

The country.cfg file is found in your main CFS2 folder. It contains information on all of the nationalities shown in your installation. The stock file contains a very minimal entry for Britain and Germany (left over from CFS1) and detailed entries for the USA and Japan. It also shows you where to begin entering new nationalities. If a country is entered into this file it will show as a nationality choice in Mission Builder as well as in the Campaign and Single Missions sections of CFS2.

For the best explanation of the country.cfg file go to Microsoft's official CFS2 website: http://www.microsoft.com/games/combatfs2/articles_SDK_countryfile.aspx. While you're there, be sure to download the **SAMPLE COUNTRY.CFG FILE.** This is the 'official' expanded country.cfg file used by just about everyone flying CFS2. This new file adds 13 new nationalities, mostly in Europe, so it is a must for those of you wanting to setup a European installation of CFS2. The new nationalities contain only the basic information and not all of the award, rank, and name information included under the USA and Japan. You can either follow the format and create your own information (especially nice for names) or look on the Internet for CFS2 nationality packs that include all of the nationality information plus award images and new screens specific to those countries.

You are limited to 19 total nationalities. If you don't like the ones in this download you are always free to add you own. Just remember, if you write a bunch of missions for the Icelandic Air Force, upload them to the Internet, and you have Iceland as number **6** in your modified country.cfg file, those missions will show up under Italy for the rest of us.

descrip.dat

The descrip.dat file is found in the INFO folder and contains a brief description of all of the objects you can place in a mission. You see these descriptions when you click the **Information** tab of any infrastructure aircraft, ship, or object that you've selected, such as the **Factory Type C** shown to the left. Open the file using Notepad and you'll see simple instructions for editing the information.

You can add your own descriptions for new add-ons you place in the game and probably do a better job than some of the stock descriptions.

DP Files

Every aircraft, ship or object used in CFS2 has a **dp** file associated with it. For ships and aircraft it is found in the main folder for that object. For everything else the dp file will be found in the **OBJECTSDP** folder within the main CFS2 directory. The dp file, or **d**amage **p**rofile, contains a wealth of information on an object and there are whole tutorials written to explain the various coding and how it works. Plus, you can download utilities for constructing or editing dp files. We will focus only on one small area of the file, the **[MISC DATA]** section**.** You will recall from Chapter Sixteen that if an aircraft is missing the [MISC DATA] section it may not show up in Mission Builder.

[MISC_DATA]
unit_family=1
category=1
allegiance=4
max_group_size=8
min_speed=240
cruise_speed=436
max_speed=654
min_alt=50
cruise_alt=6667
max_alt=10000
entered_service=9/1/43

Open the stock A6M5 Zero's dp file using Notepad and scroll down to the [MISC DATA] section and you'll see the illustration to the left.

If you ever have to add a [MISC DATA] section to an aircraft the only lines you need to add are the ones shown in **bold**. The other information is nice, but not necessary to get the aircraft to show up in Mission Builder. Make sure that you match the allegiance field to the nationality of the aircraft. This is the same number as used in the country.cfg file.

The entered service date can be important under some circumstances. When flying a campaign in CFS2 the player has the option of changing his plane from whatever was written into the mission to any other plane in his AIRCRAFT folder, provided the aircraft was available on that date. The game compares the date of the campaign mission to the

entered service date in the [MISC DATA] field and if the plane wasn't in service on that date, you can't use it.

This only applies when the player tries to change planes in the campaign. It does not affect your mission building. You can put a jet fighter in a WWI campaign when you write the mission if you want to.

Here is the best list of the families and categories that I have; I believe it to be accurate. Be sure to match these to your aircraft when copying in a [MISC DATA] field from somewhere else.

Unit_Family	**Category**
1=fighter	1=fighter
2=bomber	2=heavy bomber
3=vehicle	4=torpedo bomber
4=static object	5=light bomber
5=boat or ship	9=transport
7=carrier	10=dive bomber
	13=aircraft carrier
	14=cruisers
	15=destroyers
	16=submarines
	17=battleships
	18=patrol boat
	19=tanker
	21 - tracked vehicle
	22 - wheeled vehicle
	24=amphibious vehicles and ships

INFO Image Files

Within your INFO folder you will see a large collection of images. These are the images that display when you click the **Information** tab of any aircraft, ship, or infrastructure object that you've selected, such as the **Factory Type C** illustration shown on page 196. Personally, I find these images more helpful than the descriptions.

Many add-ons don't come with image files but you can easily create your own. From within CFS2 you can go into Chase Mode then tab to get behind any object in the game. You can then pan around to get the best view and take a screen shot using any one of the many utilities out there or by simply clicking **PrtScn** on your keyboard and pasting the image into **Paint**. The default image size is 290 x 246 pixels and I suggest that you reduce your image to that size. If your image is larger than that you may not see the whole thing in the limited Information tab window. Name your image and save it as a **jpg** file into the INFO folder.

Your image name <u>must perfectly match</u> either the **folder** name for aircraft and ships or the **dp** name for other objects. The aircraft images will only be used when placing the plane as an infrastructure object. If you click the Information tab when selecting a flying aircraft, either Player or AI, you will see a rotating view of the actual aircraft.

mdlnames.dat

This is a handy file, located in the main CFS2 directory, which seems to cause some players a lot of grief, and it shouldn't. **The primary purpose of the mdlnames.dat file is to provide a plain language name for the ship or infrastructure object being selected in Mission Builder**. It's just that simple.

Any aircraft that that has been properly installed will show up in Mission Builder without needing to be entered into the mdlnames.dat file. In fact, there is no benefit in doing so since the aircraft names displayed in the Mission Builder drop-down menu come from the title line in the plane's aircraft.cfg file and not the mdlnames.dat file.

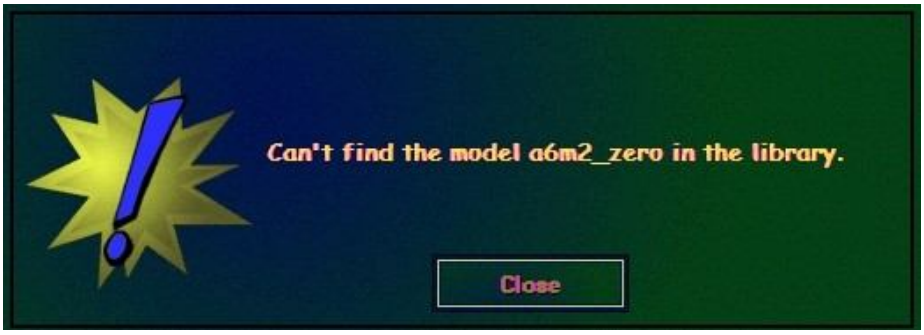
Again, the primary purpose of the mdlnames.dat file is to display a clearly understood label for the ship, vehicle, or infrastructure that you select in Mission Builder. Without a mdlnames.dat entry, Mission Builder will display either the **folder** name (ships) or **dp** name (vehicles, infrastructure) of the object depending on what kind of an object it is.

There is so much written about the mdlnames.dat file and its impact on the game that I ran some tests using a purely stock installation of CFS2; here's what I found:

1. If you remove the mdlnames.dat file completely the game will not start. You will get the "Failed to initialize..." error message.
2. If you remove all of the data from inside the mdlnames.dat file but leave the empty file in place, both the game and Mission Builder start and run normally.
3. It appears that while the game looks for a mdlnames.dat file it is not critical that the file actually contain any data for everything to still work. **Ships, vehicles, infrastructure and planes can all be added and used without any entry being made in the mdlnames.dat file**. The primary purpose of the file is to display a more meaningful name in Mission Builder.

When adding new ships or infrastructure you should edit the mdlnames.dat file so that you will see a more useful name when selecting the object in Mission Builder. Simply open the file, scroll to the bottom, and add the new information following the existing format. Be sure to place the **folder** name or **dp** file name in quotation marks, followed by an equal sign and whatever plain text you want to see in Mission Builder. After all, doesn't this: **Antiaircraft Gun (Manned) – M1 40mm – U.S**., make a whole lot more sense than this: **AF_40mm_GM**, when searching for the right piece of infrastructure?

There is a secondary function of the file that most of us have run across and one which can cause you a problem until you understand it. If you remove a stock aircraft from the game and don't remove the mdlnames.dat entry for that plane, you will get an error message when starting the game similar to this one, naming the missing aircraft. You can simply click **Close** and the game will still load.



This situation has even less effect on the Mission Builder. It will load normally without any error message; the missing aircraft simply doesn't show up in the drop-down list.

To eliminate this error message do one of two things, either delete the lines referencing the stock plane from the mdlnames.dat file, or comment that line out. To comment a line out, meaning CFS2 will skip the line and not read it, simply add a semicolon (**;**) in front of the line of text.

"A6M2_Zero"=Mitsubishi A6M2 "Zero"	**Unedited entry, CFS2 looks for this plane.**
;"A6M2_Zero"=Mitsubishi A6M2 "Zero"	**Semicolon added, CFS2 doesn't read the line, doesn't look for this plane.**

Now that you know this, you can figure out that **to remove all of the stock aircraft from the game and not get an error message at startup simply delete the entries for the stock aircraft in the mdlnames.dat file**. And that's what we do when we create additional installations of CFS2 for WWI, Europe, North Africa, Korea, Vietnam, etc. and don't want those pacific fighter planes in our game.

I've seen lots of posts at various CFS2 forums where players attribute problems to the mdlnames.dat entries or claim that a certain install won't work properly without that entry. I'm inclined to think that the actual problem lies elsewhere in their game. However, with the thousands of downloads available out there, and the literally millions of possible configurations, anything is possible. We've all experienced abnormal behavior in CFS2 from time to time. Since there is no quality control over downloads we are all at the mercy of the designer. Who's to say what potential conflicts can arise? Push a domino over here and something else may topple way over the horizon. Even so, I think the mdlnames.dat file gets blamed for a lot of problems it isn't responsible for.

opareas.dat

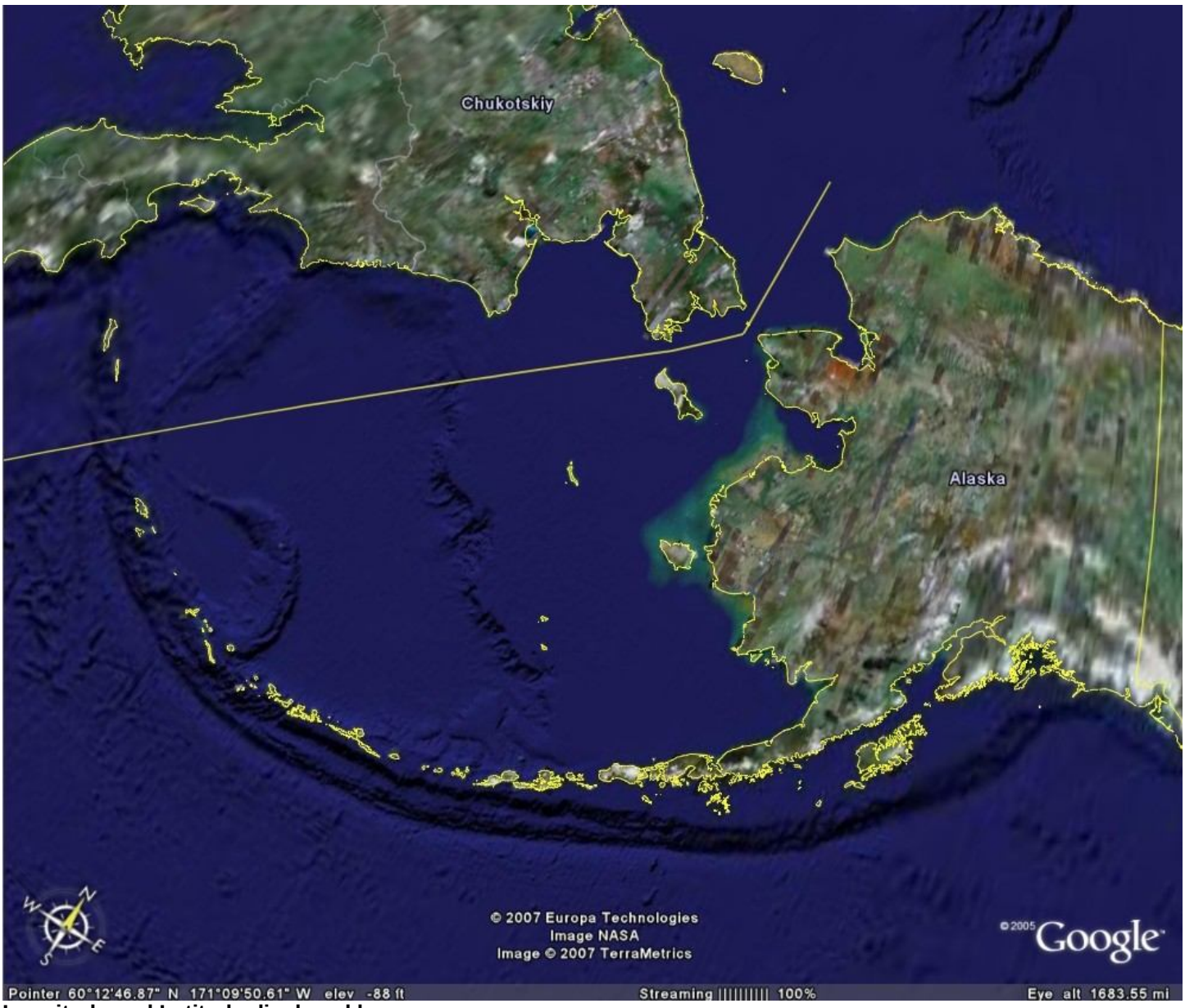
The opareas.dat file is located in the INFO folder and contains the data defining the various Operating Areas (or Theaters if you prefer) contained in CFS2. There are seventeen Operating Areas defined in the stock opareas.dat file. You can easily add more if you like.

Operating Areas within Mission Builder are used to quickly place you in a geographic area from which you can write missions. For example, if you wanted to write missions based on Pearl Harbor, begin by clicking **New Mission**. In the pop-up window click **Operating Area** then select **Hawaiian Islands** and click **Apply**. You'll now find yourself transported to the Hawaiian Islands. Without this handy feature you would have to drag the map around at the beginning of every mission or know the longitude and latitude of Hawaii and enter it in by hand each time.

When you select an Operating Area, the airbases list found under **New Mission** will display all of the airbases available for your use in that Operating Area. It will include all of those that fall directly within the limits of the area defined in the opareas.dat file as well as those within a certain radius outside of that area. For example, you can select "Midway" as an Operating Area. Although the actual Midway operating area as defined by its longitude and latitude is very small, you can select airbases in the drop down list from as far away as Port Moresby. Those bases don't fall within the actual Operating Area but are still available to you.

It appears to me, based on my testing and experience, that there is an area surrounding the center of an Operating Area that extends out about 3,800 to 4,000 statute miles and that any airbases falling within that radius will appear on the drop down list for you to use. I can't tell if it is an imaginary circle drawn around the center of the Operating Area or a box based on longitude and latitude. Whatever it is, as you move around the world from Operating Area to Operating Area you will see airbases drop off the availability list and new ones appear. This is obvious to you when you select **European Theater** as your Operating Area. You will no longer see any of the Pacific bases on the airbases list but will see all of the old CFS1 European bases listed.

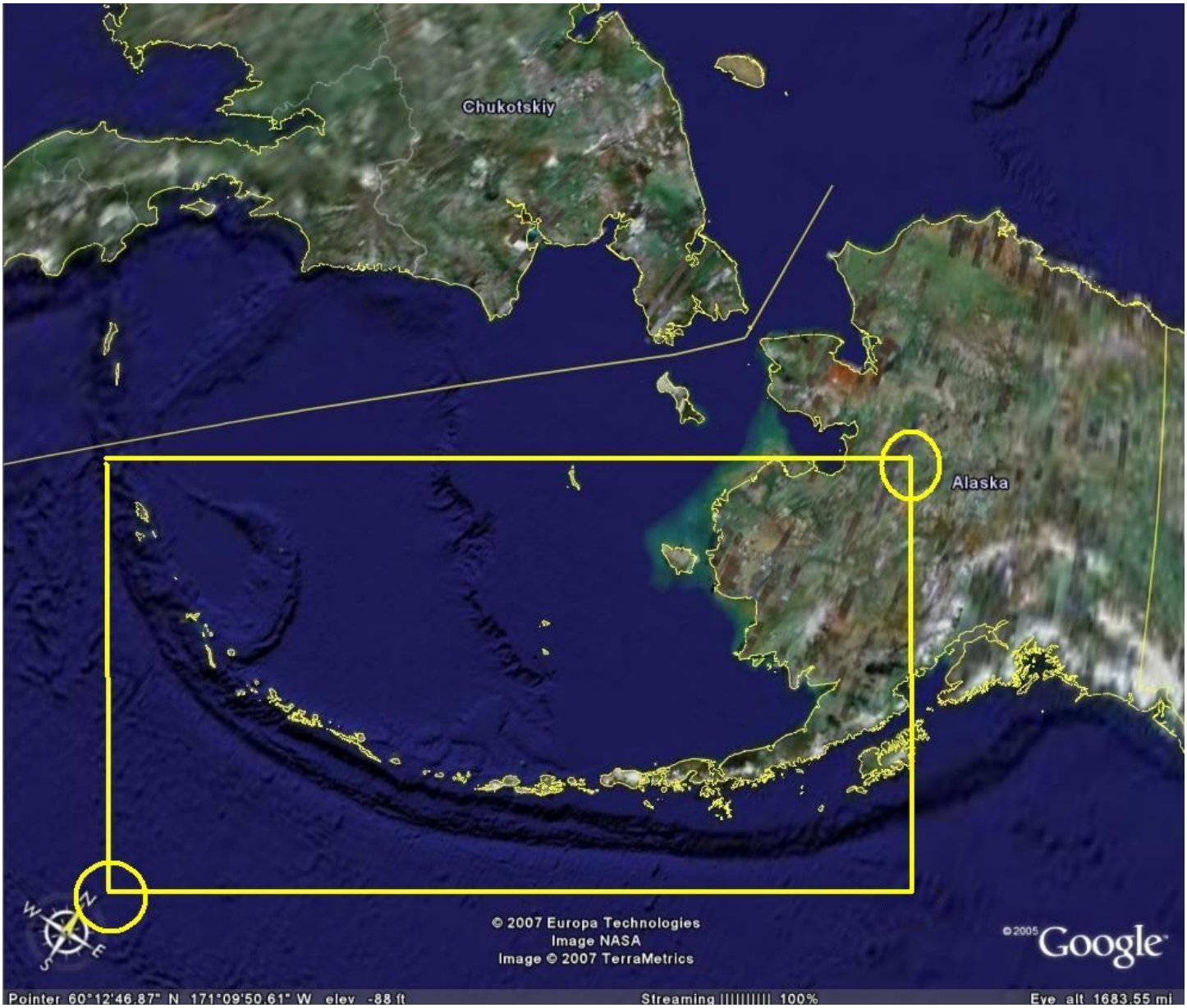
Adding new Operating Areas is easy. Picture a world map hanging on your wall. An Operating Area is a box drawn around a portion of the map. To make your entry you need to define two opposite corners of the box, either Northwest/Southeast or Southwest/Northeast. Find these two spots and write down the longitude and latitude of each. You can use a good map, a globe, or better yet, **Google® Earth**. Let's say we wanted to add an Operating Area for the Aleutian Islands. First zoom in on the Aleutians using Google® Earth. You might see an image similar to this:



Longitude and Latitude displayed here

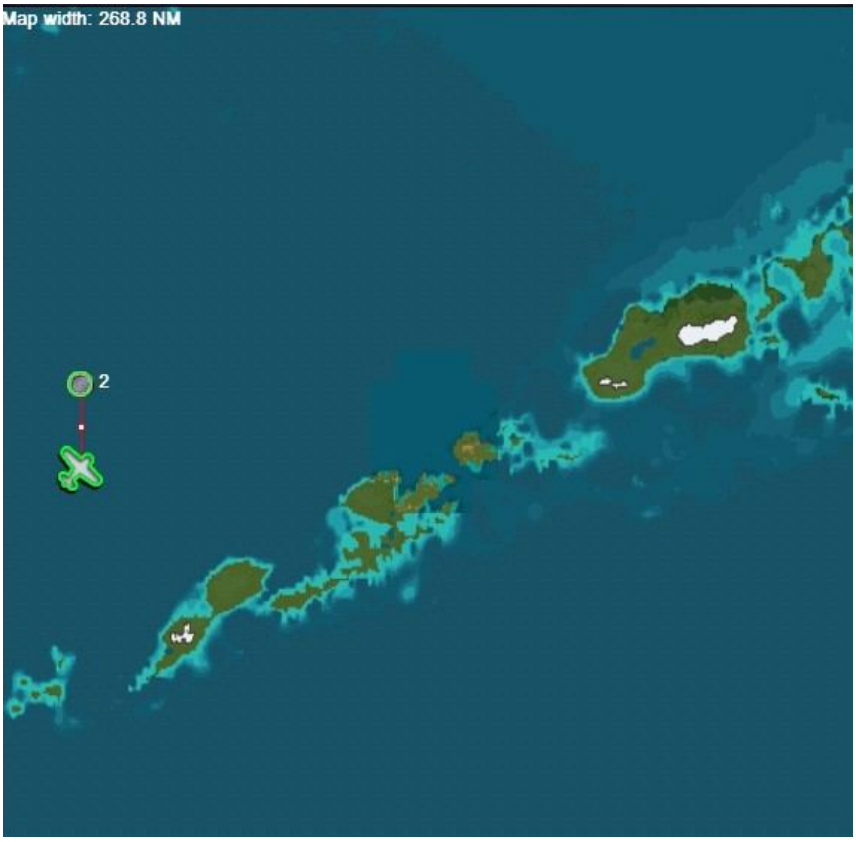
Note that the longitude and latitude of your cursor is displayed in the lower left hand corner of the image and changes as you move the cursor around.

Now picture a box drawn around the Aleutians, this will be our Operating Area. We need the exact longitude and latitude of two opposite corners of the box. In this illustration I have circled the southwest and northeast corners. Move your cursor over these two spots and write down the longitude and latitude displayed on the screen.



My locations came out as **N44 06, E178 48,** (southwest corner) and **N64 23, W157 41** (northeast corner). Open the opareas.dat file and study the format. To add our new Operating Area we simply follow the existing format and add our new information to the bottom, giving it a name and inserting the longitude and latitude that we noted. Our new line would look like this:

Aleutian Islands, N44 06, E178 48, N64 23, W157 41



Save the file, start up Mission Builder, select your new Operating Area, and click **Apply**. When the map loads, there you are sitting in the Aleutians instead of The Slot.

Operating Areas are a handy way to jump around the CFS2 world as your mission building expands to other theaters of the war. Two or more Operating Areas can overlap without any problem. You can make your Operating Area any size provided you do not cover more than 90 degrees of longitude or latitude on a side.

If you like the idea of additional Operating Areas but don't want to create your own, there are several players out there who have made excellent custom opareas.dat files available for download.

outcome.dat

The outcome.dat file is located inside your INFO folder and contains the text messages that appear after a mission when you've been shot down, get promoted, are wounded, killed in action, sent home, etc. The format is pretty straight forward. You can change the wording of any of the text if you choose to do so. Each category of message has multiple options. Which message is selected for display is defined by some process buried in the CFS2 code.

sculttlebutt.dat

This file is also located in the INFO folder and contains the scuttlebutt messages that appear as a part of the stock campaigns. The file starts with a good explanation of the format used. You have several different options here. You could easily edit the scuttlebutt to something else if you'd like. If you are really ambitious, you could create entirely new scuttlebutt for different nationalities and insert them in a campaign written for those countries.

Readme

Here's a novel idea; when all else fails read the instructions. CFS2 contains a Readme Word document located in the main directory. Take time to read through it. It contains some very interesting information regarding multiplayer, mission building, add-ons, scenery, etc. Even the Global Scenery Layer is covered here.

Chapter Nineteen – Mission Accomplished

Debriefing
Congratulations! You've come a long way since that first time you fired up Mission Builder and stared at that empty map wondering, "What do I do next"? Hopefully you've enjoyed the journey, I have. The tips, tricks, and techniques covered in this Handbook are intended to help you master the art of mission building and get many more years of enjoyment from CFS2. Don't hesitate to experiment on your own and see how far you can push the Mission Builder envelope.

Be creative in your mission design. Not every mission has to be a gunfight. You can create some very interesting and challenging missions by going beyond basic air-to-air combat. For example:

- Create a Search & Rescue mission using a PBY. Set an infrastructure PT boat on the ocean, add a smoke marker to it to simulate a fire and have your PBY fly to the spot to perform a rescue. You can create an event area around the PT boat and require the player to land and taxi close to the boat to effect the rescue. Hold the player there while the crew comes aboard then instruct your player to take off and return to your base.
- Set up a mission at night in a thunderstorm with visibility reduced to one mile. Place your player 5-10 miles from the base. Instruct him to reduce his fuel load to 5% and turn off his Tactical Display. Then see if he can find the base and land safely using only his compass before his fuel runs out. Expect lots of crashes.
- Trigger an engine fire at some point and require the player to make an emergency, wheels-up, landing. Use triggers and events to make the belly landing a mission goal.
- Be creative – use our imagination!

Although we've covered a lot of ground, this Handbook isn't the last word in mission building, nor are the techniques taught here the only way to do things. In fact, you'll learn that there are many different ways to accomplish the same thing in CFS2. If you have a favorite technique that's different from what we covered here, but gets the job done, use it. You only need to ask one question at any of the forums to find out how many different opinions there are on how to do things in CFS2. The methods that I have demonstrated here have been time tested and work for me. Over the years I have found them to be the most reliable and/or easiest way to accomplish a task. That doesn't mean they will be the easiest for you.

Remember too, all of my data is based on a stock installation of CFS2. That is the only common denominator we all have to work with. As your installation grows and you modify essential files, you may find that your game and the Mission Builder start to behave differently than the examples I have used here. I can't explain those one-of-a-kind occurrences. I know from my own heavily customized installations that sometimes unexplained things just happen. I simply scratch my head and move on.

Improving AI Performance
Although it isn't directly related to mission building, I guess no discussion of CFS2 would be complete without mentioning the strange, if not down right abysmal, performance of the AI aircraft from time to time, both your own wingmen and the enemy flights. I'm sure at one time or another you've experienced situations like these:

- You write a mission where your flight or another AI flight is to land; some do, some don't, some land then overshoot the runway by a half mile or more.
- Your wingmen fly around and around seemingly oblivious to the enemy planes or ground targets leaving you to do everything.

- Your wingmen seem to fly circles around the enemy fighters without shooting.
- Whether you're in a flight of 8 or 80 **you** are the center of attention. Every AA gun seems to be trained on you and all of the enemy fighters pursue you, even to the point of ignoring your wingmen or flying through them to attack you.
- AI planes crash on take off, landing, turning, or for no apparent reason.

Unfortunately, poor AI performance has been a recurring theme among CFS2 players since the beginning. To date no one has found a perfect solution, although there continues to be a lot of effort being put forth to find answers. Some of the work is rather arcane, all of it is interesting. In time someone may crack the code and we will see real improvement. As a mission builder there are only a few things that you can do to try and improve things.

1. Set all of your wingmen to skill level **Ace** and aggression level **High**. I've found that anything less and you get very unreliable performance. Even at these settings it is no guarantee of good performance, but it will improve your odds that they will hit what they are aiming at and attack their targets.
2. Don't start in the air at too low of an altitude over ground. If your AI are crashing at mission start it may be because they are flying into the ground. This can be a real problem if you have added any custom mesh scenery. Try setting your starting altitude higher.
3. Avoid sharp turns. AI formations don't like sharp turns. Large bombers seem to have the most trouble, crashing into one another when turning. You can reduce this problem by plotting wider turns. Use more waypoints, just like we did with ships in Chapter Nine, when plotting turns.
4. Use Front Lines to clearly differentiate friendly from enemy territory and target specific objects in Strike missions.
5. AI level bombers do not track moving ships the way torpedo bombers and dive bombers do. They will drop their bombs on the spot you mark on the map. As a result, hitting a moving ship with a level bomber is very difficult. Either use lots of bombers and carpet bomb the location or stick with torpedo or dive bombers when attacking moving ships.
6. Unfortunately, many add-on AI aircraft behave less reliably than do the stock planes. Still, I would never suggest not downloading and using them. Just be aware that sometimes add-ons are tougher to compensate for.

In the game use your wingmen commands. Split your flight when approaching a target. Use the yellow brackets to track the target then give your wingmen the **Attack** command. If you split your flight, wait for their response ("Roger" or whatever) then bracket a second target and give another Attack command. This will send the other wingmen after the second target. With a large flight, such as seven wingmen, you'll still have someone to follow you toward a third target.

If you get in trouble use the **Recall** command, bracket the enemy plane chasing you and use the **Attack** and **Help** commands. Fly toward your wingmen, try to get them to engage the enemy. Remember, AI aircraft have unlimited fuel and unlimited ammo so you can't dodge them until they run out of bullets and hope to escape. However, if you aren't too badly shot up you can almost always outrun them, and after a while they will break off and give up the chase.

Spend some time at the various CFS2 forums and see what tricks other players are trying to get the AI to perform better. Some of these techniques involve hand-editing various files, but after reading Part III, you're probably comfortable with that by now. In the end, the best recommendation I have is to be patient. Time and again I have seen AI planes fly around and around but eventually turn and strike their target. Even if you run out of ammo, your wingmen haven't. Give them time and they are likely to finish the job.

Every once in a while things go exactly according to plan. I have an anti-ship mission that I fly with two wingmen. We approach three Japanese torpedo boats from the rear. As we near, I give the split command; bracket the boat on the left and give the attack command, then repeat the procedure for the boat on the right. I take the boat in the center. We dive on our targets, firing guns and rockets. In one pass all three targets are destroyed. Sometimes the only ship not sunk is mine! It's a mission that works perfectly every time; and it gives me hope for the future.

Acknowledgements
First, I'd like to thank the many simmers who've downloaded the original version of this Handbook and whose questions, encouragement, and emails have kept me motivated to keep pushing the mission building envelope. I especially want to thank Martin Wright, whose utilities have enhanced the CFS2 experience beyond measure, and who has graciously taken time to answer my emails and help me work through some oddball problems. I also want to give a tip of the hat to Chris Westervelt (Maskrider). His Pacific airbases turned CFS2 on its ear. He raised the bar for what we expect airbases to look like and reinvigorated an aging flight sim. I'd like to thank Microsoft® for developing Combat Flight Simulator 2 and giving some of us a chance to recapture our youth. Most especially I thank my wife, who lovingly tolerates my cyberlife and graciously consented to edit this work, and, of course, the real Cody Coyote who lets me use his name and photograph and who patiently sleeps at my feet as I fly the virtual skies over the South Pacific.

Legal
This electronic publication is released as Freeware. Copyright© 2007, 2011 by Jim Borowski (Cody Coyote). All rights reserved. As freeware you are permitted to distribute this digital archive without modification to its contents. No charge may be made for this electronic archive and the author's rights and wishes concerning this archive must be respected. I ask only that credit be given to the author.

☛ Well, that's the end of the road. What follows is a series of appendices that contain some additional information I hope you will find useful.

I welcome questions and comments regarding this Handbook. I hang out at several CFS2 forums, especially those at Sim-Outhouse, Simviation, and Lindsey Watt's site. I don't post too often but I read everything related to CFS2. If you have a question for me you can try one of those boards or send me an email at cody_coyote2000@hotmail.com. I make it a point to answer all of my emails.

Have fun and watch your six. It's time for me to take my master for a walk!

Cody Coyote

Appendix A

Files Included with this Handbook

Included in the main Zip file are mission files for each of the missions we wrote in this Handbook as well as a mission referenced in Chapter Fifteen – *How to Take Off from any Point on the Planet*. You can fly these missions in CFS2 to see how they are supposed to work and then open them in Mission Builder to see how they are constructed. I would recommend that you not attempt to modify them as they would lose their value as reference material. The name of each mission is preceded with an asterisk so that it will appear at the top of your Single Missions list making them easier to locate.

Also included is a blank cmg file and two mission files stored as PDF documents. These are a flight of 8 Wildcats and one of 16. I chose this format so that the new data for the flight of 16 could be highlighted. You could copy the mission into Notepad and save it as a mis file if you wanted to fly it. I'd suggest you write your own, it's more fun.

The files included with the Handbook are as follows:

File name	Mission name	Chapter reference
CC_HB_01.mis & dyn	*01 Beginner's Luck	Chapter Two
CC_HB_02.mis & dyn	*02 Raid on Tarawa	Chapter Three
CC_HB_03.mis & dyn	*03 A "Wake" Up Call	Chapter Four
CC_HB_04.mis & dyn	*04 Luck of the Draw	Chapter Five
CC_HB_05.mis & dyn	*05 Alternate Reality	Chapter Six
CC_HB_06.mis & dyn	*06 A Brief Hop Around Iron Bottom Sound	Chapter Seven
CC_HB_07.mis & dyn	*07 No Way!	Chapter Eight
CC_HB_08.mis & dyn	*08 Flying with Friends	Chapter Fourteen
CC_HB_09.mis & dyn	*09 Henderson Alternate Take Off	Chapter Fifteen
CC_HB_TEMPLATE.cmg	Blank campaign file template	Chapter Seventeen
Modified Mission - Flight of 8.pdf	Flight of 8 Wildcats	Chapter Fourteen
Modified Mission - Flight of 16.pdf	*1 Numbers Test	Chapter Fourteen

Appendix B

Transcripts of Radio Transmissions used in Mission Builder

The following is a transcript of the English language radio transmissions available for use in Mission Builder. Of the 563 sounds available to you, the first 270 are in Japanese. I have not translated those. However, based on a random sample of about a dozen of these Japanese transmissions they appear to match their English counterparts. In other words, transmission jtr033 appears to say the same thing in Japanese that tr033 says in English.

This list is intended to help you quickly find and add a sound in Mission Builder. Simply scan through the list to find the appropriate radio transmission. The number in the left-hand column matches the file name shown in Mission Builder. You will notice that some sounds are repeated. The differences are in the tone of voice used in each transmission

t037 "All right you guys, get those planes down. We'll have to send out another strike."

t038 "Nice try, another strike is ready to launch."

t039 "Home plate calling flight leader. Stand by for new mission orders.

t040 "Home plate to flight leader. Return to base immediately."

t041 "Home plate to flight leader. Primary target is too heavily defended. Switch to your secondary target. Repeat, switch to your secondary target."

t042 "Base to flight leader. Coast watchers report enemy aircraft approaching from the north at high altitude."

t043 "Base to flight leader. Coast watchers report enemy aircraft approaching from the east at high altitude."

t044 "Base to flight leader. Coast watchers report enemy aircraft approaching from the west at high altitude."

t045 "Base to flight leader. Coast watchers report enemy aircraft approaching from the south at high altitude."

t046 "Base to flight leader. Coast watchers report enemy aircraft approaching from the north at medium altitude."

t047 "Base to flight leader. Coast watchers report enemy aircraft approaching from the east at medium altitude."

t048 "Base to flight leader. Coast watchers report enemy aircraft approaching from the west at medium altitude."

t049 "Base to flight leader. Coast watchers report enemy aircraft approaching from the south at medium altitude."

tr001 “All right, in this flight we’ll work on taxiing and takeoff procedure. For taxiing more movement on the controls is necessary for maneuvering than when we’re in the air. The rudder is the most important control in taxiing. You won’t need to taxi for this takeoff because you’re already lined up on the runway. All right let’s go ahead and start the plane but keep the brakes on so we don’t move. Press “E” on your keyboard to start the engine.”

tr002 “Press “E” on your keyboard to start the engine.”

tr003 “Let’s go, kid I don’t have all day.”

tr004 “Anytime today would be good.”

tr005 “Remember to start at the end of the runway so that you have the full length for the takeoff run. You can’t use the runway behind you. First, extend the flaps one notch to give it more lift.”

tr006 “If you don’t know where the flaps are check the manual.”

tr007 “Drop the flaps one notch.”

tr008 “While we’re young cadet, I don’t have all day.”

tr009 “OK, let’s give it a shot. Release brakes and slowly give her full throttle keeping her pointed down the runway until she leaves the ground.”

tr010 “Let’s go cadet, release the brakes.”

tr011 “You’re climbing too fast. Ease the stick forward a little.”

tr012 “Ease the stick back a little.”

tr013 “You’re too slow. Increase power.”

tr014 “You’re off the runway. This lesson is over!”

tr015 “Watch your heading.”

tr016 “Well you’re off the ground. Alright now, raise your gear and flaps and let the plane accelerate until you’re at cruising speed.”

tr017 “Raise the gear cadet.”

tr018 “Retract the flaps.”

tr019 “You’re diving, pull up.”

tr020 “The nose is too high. Push forward on the stick.”

tr021 “More throttle, speed up,”

tr022 “You’ve wandered way off heading. This lesson is over.”

tr023 “OK, now that we’ve got the speed and your gear and flaps are up, let’s have you start a normal climb to 500 feet.”

tr024 “Climbing too fast. Push forward on the stick.”

tr025 “Keep those wings level mister.”

tr026 “Keep the plane flying straight.”

tr027 “You’re diving.”

tr028 “Looks like you want to wander all over the sky today. This lesson is over. You’re on your own.”

tr029 “OK, you’re at altitude now, well done. The lesson’s over feel free to fly around and get a feel for the plane.”

tr030 “What the heck are you shooting at.”

tr031 “Get off the deck. You break that plane you buy it!”

tr032 “There’s a gas shortage kid, better get that plane back on the deck.”

tr033 “Flying like that will make you lose your government issued lunch.”

tr034 “Hey, no buzzing the field. I got your number mister.”

tr035 “Where the heck are you flying to? Bring that aircraft back here right now.”

tr036 “All right, in this lesson you’ll make a landing. You should have studied your manual and lesson briefing on the approach, key position and descent. You want to be at 500 feet and facing the airfield. Now, put down the gear and extend the flaps one notch.”

tr038 “Your gear is still up.”

tr039 “Flaps have not been extended.”

tr040 “You’re exceeding the assigned altitude.”

tr041 “You’ve got to get the aircraft to 500 feet pointing down the middle of the landing airfield.”

tr042 “You’re way off course. The lesson’s over.”

tr043 “Now decrease throttle and let the plane descend until you’re 50 feet above the runway. Keep the speed constant. Let the plane descend until you’re about 20 feet off the ground in the half of the runway closest to you. Increase throttle to gain altitude, decrease to let descend.”

tr044 “You’re too high.”

tr045 “You’re too low.”

tr046 “Ease off the throttle to slow down.”

tr047 “You’re off heading. Keep the plane pointing down the center line of the runway.”

tr048 “Keep your wings level.”

tr049 “You’re all over the map. I’m ending the lesson.”

tr050 “OK, bring her down and apply brakes until the plane is stopped. Keep the plane down the middle of the runway.”

tr051 “You’re running out of runway.”

tr052 “Ease off throttle to slow down.”

tr053 “Apply the brakes.”

tr054 “Stay on course.”

tr055 “Wings level.”

tr056 ‘You’re off the runway.”

tr057 “Go around. You’re on your own.”

tr058 “Well done.”

tr059 “Nice landing.”

tr060 “Good job. You fly before?”

tr061 “Another one for the scrap heap.”

tr062 “Let’s get another cadet in, get this one out.”

tr063 “That’s gonna cost you.”

t064 “Close enough for government work.”

tr065 “You went off the runway. Listen up next time.”

tr066 “You’re off the runway. You don’t want to do that on a carrier.”

tr067 “Fly a loop, set power for cruise speed. Then pull the stick back smoothly, but firmly, to start climbing. Alright, to start this nose down and adjust the throttle until you’re at cruise speed.”

tr068 “Watch your heading.”

tr069 “Keep the plane level.”

tr070 “You’re diving. Pull up.”

tr071 “Too steep. Push forward on the stick.”

tr072 “More throttle, speed up.”

tr073 “You’re completely off course, you’re on your own.”

tr074 “Now gradually increase back pressure on the stick – stick back. Alright now, let the plane climb straight up.”

tr075 “Keep the nose up.”

tr076 “Speed up.”

tr077 “Keep the wings level.”

tr078 “Keep working the stick back.”

tr079 “Where you going? The lesson’s over.”

tr080 “When you’re almost upside down tilt your head back so you can see the horizon and make sure that your wings are still level. Keep back pressure on the stick.”

tr081 “Now, relax the back pressure on the stick as you go over the top and get the nose of the plane pointing back down.”

tr082 “You’re pulling back on the stick a bit too hard. Relax the back pressure a little.”

tr083 “More throttle, speed up.”

tr084 “Keep the wings level.”

tr085 “Keep working the stick back.”

tr086 “Where are you going? Ahhh, the lesson’s over.”

tr087 “All right, now as the nose drops down through the horizon, retard the throttle to cruising and begin to pull out until the plane is level.”

tr088 “Watch your altitude.”

tr089 “Keep the wings level.”

tr090 “Get the plane level.”

tr091 “You’re way off. The lesson’s over.”

tr092 “As the nose comes to the horizon apply throttle and gain as much altitude as you can before returning to cruise speed. Well done.”

tr093 “The Immelmann is a half loop with a half roll on top. It’s a good way to get altitude and change direction at the same time. Now start an Immelmann just like a loop, nose down and adjust the throttle until you’re at cruising airspeed.”

tr094 “Watch your heading.”

tr095 “Keep the plane level.”

tr096 “You’re diving, pull up.”

tr097 “Don’t pull back so hard.”

tr098 “More throttle, speed up.”

tr099 “You’re completely off course! This lesson is over.”

tr100 “Now gradually increase back pressure on the stick, straight back. Get the plane going straight up.”

tr101 “Keep your airspeed up.”

tr102 “Get the nose pointing straight up.”

tr103 “Watch your airspeed.”

tr104 “Keep working the stick back.”

tr105 “Where you going. The lesson’s over.”

tr106 “Now, when you can no longer see the horizon tilt your head backwards so that see the opposite horizon as soon as possible.”

tr107 “Keep the nose up.”

tr108 “More throttle, speed up.”

tr109 “Keep working the stick back.”

tr110 “Where are you going? The lesson’s over!”

tr111 “As you go over the top you’ll probably need a little more back stick.”

tr112 “Keep your airspeed up.”

tr113 ”More throttle – speed up.”

tr114 “Keep the wings level.”

tr115 “Keep working the stick back.”

tr116 “Where are you going? The lesson’s over!”

tr117 “Now roll upright. Move the stick all the way to the left or right to start the roll. As you finish the roll, use forward or back pressure on the stick to keep the nose on the horizon.”

tr118 “Watch your heading.”

tr119 “Keep the wings level.”

tr120 “You’re diving, get the plane level.”

tr121 “You’re climbing, get the plane level.”

tr122 “You’re completely off course. This lesson is over!”

tr123 “As the roll is nearing the upright position, ease the stick in the opposite direction of the roll so that the wings end up level. Good job.”

tr124 “A Split-S is the opposite of an Immelmann. It’s a quick way to convert altitude into airspeed and change direction. To fly a Split-S, make sure you have plenty of altitude and that your airspeed is below cruising. Ready? Let’s begin. At normal cruise, pull the nose up and half roll until you’re inverted.”

tr125 “Watch your heading.”

tr126 “Move the stick to the left or right to roll the plane inverted.”

tr127 “You’re diving. Get the plane level.”

tr128 “You’re climbing.. get the plane level.”

tr129 “You’re completely off course.. this lesson is over!”

tr130 “Now reduce throttle and apply back-pressure on the stick in order to dive under, with the plane pointed directly down.”

tr131 “Watch your airspeed.”

tr132 “Keep your wings level.”

tr133 “Keep working the stick back.”

tr134 “Where are you going? The lesson’s over!”

tr135 “Coordinate throttle and try to assume straight-and-level flight with a minimum loss of altitude. Pull back until you’re level.”

tr136 “Watch your altitude.”

tr137 “Keep the wings level.”

tr138 “Get the plane level.”

tr139 “Where are you going? The lesson’s over!”

tr140 “Now you should be at level flight one-hundred-and-eighty degrees from your original heading.”

tr141 “The high side attack requires deflection shooting. The idea is to set up the shot on the target from an angle above and to the side of the enemy, and to shoot in front of the enemy’s path as you converge. Head towards the enemy plane, but stay on your current heading of zero-zero-zero.”

tr142 “Keep your heading.”

tr143 “You’re off your heading.”

tr144 “Get back to your original heading.”

tr145 “Where are you going? The lesson’s over!”

tr146 “Now turn in to the target, but fly to a point ahead of him, instead of straight at him.”

tr147 “Keep in front of him.”

tr148 “You’re off your heading.”

tr149 “You’ve gone past him.”

tr150 “You’re all over the map – try again.”

tr151 “Fire where he’s GOING to be and let him fly right into the gunfire.”

tr152 “Fire!”

tr153 “Shoot in front of him!”

tr154 “Get closer before you fire.”

tr155 “Hurry!”

tr156 “You missed – try it again.”

tr157 “Got him – well done!”

tr158 “The overhead attack commences from dead ahead and at least two thousand feet above the target, on an opposing course. As you attack, roll the plane on its back and then pull through into a vertical dive. Let’s do it. Hold your heading and altitude and wait until the target passes under you.”

tr159 “Hold your heading.”

tr160 “You’re off course.”

tr161 “Wait until he’s under you.”

tr162 “Where are you going? The lesson’s over!”

tr163 “Move the stick to the left or right to roll the plane inverted.”

tr164 “Watch your heading.”

tr165 “You need to be inverted!”

tr166 “You’re diving.. get the plane level.”

tr167 “You’re climbing.. get the plane level.”

tr168 “You’re completely off course, this lesson is over!”

tr169 “Pull through into a dive towards the target, adjusting to keep your sight ahead of it.”

tr170 “Keep in front of him.”

tr171 “You’re off course.”

tr172 “You’ve gone past him.”

tr173 “You’re all over the map – try again.”

tr174 “Fire when range is one thousand feet or less – remember to lead him”

tr175 “Fire!”

tr176 “Shoot in front of him!”

tr177 “You’re firing too far away.”

tr178 “Hurry!”

tr179 “Now pull up and turn to break away from the target, with a smooth pull-out to convert excess speed back into altitude. Nice job”

tr180 “You missed – try it again.”

tr181 “Dive bombing is just what its name infers – diving your plane towards the earth at sixty-plus degrees, and releasing your bomb at the target in this dive. Let’s start the run. Start by dropping down toward the target in a steady accelerating glide.”

tr182 “Head towards the target.”

tr183 “You’re off course.”

tr184 “You’re diving too fast.”

tr185 “You’re way off course – the lesson’s over”

tr187 “Watch your heading.”

tr188 “Get the plane inverted .”

tr189 “Roll the plane on its back.”

tr190 “Get the plane on its back.”

tr191 “You’ve passed the target – try again.”

tr192 “Now get the plane diving almost vertically. Use the stick and rudder to keep the bombsight on the target. Remember – the more vertical you are, the less the bomb will horizontally “drift.”

tr193 “Keep the crosshairs on the target.”

tr194 “Head straight towards the target.”

tr195 “You’ve lost the target.”

tr196 “Pull up and try again.”

tr197 “Release the bomb on target.”

tr198 “Release!”

tr199 “Watch the target!”

tr200 “Altitude!”

tr201 “Pull up and try again.”

tr202 “Now pull out and bank so that you can see if the bomb hit the target.”

tr203 “Bingo!”

tr204 “Close but no cigar. Try again.”

tr205 “The trick to strafing is to stay low and fast and to stay on the deck after you’ve made your run. OK, let’s give it a try. Turn in at least one thousand yards from your target and dive toward it.”

tr206 “Head towards the target.”

tr207 “You’re off course.”

tr208 “Find the target.”

tr209 “You’re way off course – the lesson’s over.”

tr210 “Get to the deck as fast as possible, keeping your eye on the target. Remember, as long as you’re in the air, you can’t get too low on a strafing run!”

tr211 “Watch your heading.”

tr212 “Get lower.”

tr213 “Head at the target.”

tr214 “Watch your airspeed.”

tr215 “When the target is within 1,000 yards, dip your nose down and open fire. Start shooting before the target, and let the fire go through the target as you fly by.”

tr216 “Fire!”

tr217 “Keep firing!”

tr218 “Walk that fire in!”

tr219 “Pull up and try again.”

tr220 “Bingo!”

tr221 “Try again.”

tr222 “Once you clear, stay low! Don’t climb! They’ll still be shooting at you. Stay low and fast and clear the enemy’s gunfire. Don’t worry about looking back to see if you’ve hit your target – get clear first.”

tr223 “All your squadron mates depend on your ability to find the ship and get aboard promptly. Follow your checklist and make sure you’re lined up when you reach the deck. When you’re in the groove, the LSO will guide you down.”

tr224 “Too high”

tr225 “Drop your hook!”

tr226 “Lower the gear.”

tr227 “Extend your flaps.”

tr228 “You’re off course.”

tr229 “Too low.”

tr230 “Ease off throttle and slow down.”

tr231 “When the LSO gives you the “Cut!” command, close throttle immediately. Chop the power, let the nose start down, then pull back on the stick to make a three-point landing. Remember, don’t dive for the deck!”

tr232 “OK, So you’ve decided to be nice about it and let the Landing Signal Officer help you get aboard? So you’re aboard, hotshot!”

tr233 “Someone get me a refill of coffee.”

tr234 “Go around!”

tr235 “For combat efficiency against enemy forces on recent missions, you have been given an award.”

tr236 “For conspicuous service in operations against the enemy you have been awarded a commendation.”

tr237 “For meritorious service against the enemy, you will receive an award.”

tr238 “For distinguished and meritorious performance of duty while participating in aerial flight against the enemy, you have been given a commendation.”

tr239 “Congratulations! Your skill and guts have paid off. You’ve been promoted. Keep up the good work!”

tr240 “For your hard work, aggression and devotion to duty, you’ve been promoted.”

tr241 “Looks like you’re up for a promotion – well done!”

tr242 “It looks like you’re eligible for a promotion.”

tr243 “You’ve been doing so well, you’re up for a promotion.”

tr244 “The Secretary of War regrets to inform you that your loved one has been killed in action.”

tr245 “It is with deep regret that the Secretary of War must inform you of the death of your loved one.”

tr246 “I’m sorry to say that your loved one has been killed in battle. Please accept my heartfelt sympathy for you in your great sorrow and loss.”

w636 “Whoa! That blew up real good!”

w637 “Looks like our carrier has had it.”

w638 “Our carrier has taken hits”

w639 “We’re under attack! Get back here as fast as you can!”

w640 “Scratch one Flattop!”

w641 “Let’s follow this guy back to his ship.”

w642 “Scramble! Scramble! Enemy fighters inbound!”

w643 “What are you waiting for, get moving!”

w644 “He’s dropping his cargo.”

w645 “Looks like they’re puttin’ steel on target.”

w646 “Maybe we should head east, sir”

w647 “Maybe we should head west, sir”

w648 “Maybe we should head north sir”

w649 “Maybe we should head south, sir”

w650 “Just a walk in the park!”

w651 “Bull’s-eye!”

w652 “Nothing here. What about reports from our recon planes”

w653 “Photo mission complete.”

w654 “Ground Leader to Flight Leader, we’re under attack. Need help!”

w655 “We can follow that patrol boat’s vector.

w656 “I missed my eggs and ham.”

w658 “Tag – you’re it!”

w659 “My fuel’s leaking. I have to head back.”

w660 “Heavies above.”

w661 “Heavies below.”

w662 “That’s one for the record book.”

w663 “We’re getting the hell pounded out of us!”

w664 “Torpedo Leader to Flight Leader. We need your cover as we make our run.”

w665 “Bombing Leader to Flight Leader. We are commencing our attack.”

w666 “Ground Leader to Flight Leader. Enemy troops to our West. You are cleared to attack.”

w667 “Ground Leader to Flight Leader. Enemy troops to our South. You are cleared to attack.”

w668 “Ground Leader to Flight Leader. Enemy troops to our East. You are cleared to attack.”

w669 “Ground Leader to Flight Leader. Enemy troops to our North. You are cleared to attack.”

w670 “Ground Leader to Flight Leader. Cease-Fire. Cease-Fire. You’re hitting friendly troops!”

w671 “Bomber Leader to Flight Leader. Drop Complete, we’re heading home.”

w672 “Torpedo Leader to Flight Leader. All Fish in the water, we’re heading home.”

Appendix C

Ten Downloads I Wouldn't be Without

While there are hundreds, if not thousands, of great CFS2 add-ons and downloads out there, here are ten I would not be without when it comes to maximizing my Combat Flight Simulator 2 experience.

ACMan Aircraft Manager by Martin Wright (http://fly.to/mwgfx/). This handy little tool allows you to move aircraft quickly and easily in and out of your AIRCRAFT folder to keep your total plane count under 100. You can create "hangers" (special folders) to store your extra aircraft by nationality or what ever other criteria you choose to use. And, you can drop a version into each of your CFS2 installs and have it work independently of the others by renaming it to something like ACMan_PTO, ACMan_ETO, etc. This is a must utility if you plan on adding lots of additional planes to CFS2.

MKAfd Martin Wright's (http://fly.to/mwgfx/) AFD and Runway Generator is simply the quickest and easiest way to create a new runway for CFS2. It won't add infrastructure or ground textures, but if you're looking for a quick way to drop a permanent runway into your game and want to build up the surrounding area on your own, this is the tool.

Mgrab Martin Wright's (http://fly.to/mwgfx/) Multiple Screengrab Utility. There are lots of ways to grab a screenshot off of CFS2, simply pressing PrtScn and pasting the image into Paint will work. But if you're looking for a quality utility that allows you to capture multiple images one after another then I recommend Mgrab. Some player's like another free alternative, GrabClipSave. Both are excellent, I just prefer Martin's track record (can you tell I'm a Martin Wright fan?).

DXTBmp Another must-have Martin Wright (http://fly.to/mwgfx/) utility, his DXT and 16/32 bit Texture Manipulator. If you are even remotely interested in messing with textures, whether you plan to create an entirely new skin or just add kill flags below the cockpit, you must have this tool to derive the best results. No other single texture tool can do so much so easily.

SplitGVR Martin Wright's (http://fly.to/mwgfx/) utility for opening the GVR sound files and extracting the individual radio transmissions. This is the only tool of its kind and essential if you plan to do the types of modifications we talked about in Chapter Twelve.

CFO Weather This is a unique, multi-function, utility invented by Bill Potvin II, one of the early geniuses of CFS2. It loads with the game and is added to the tool bar at the top of the screen. With it you can do all sorts of things such as control the simulation rate (way beyond the few options normally obtained with the keyboard), jump directly from within a flight to other locations such as Free Flight, Quick Combat, etc.; even directly into Mission Builder. It allows you to set certain "cheats" while flying, has an auto pilot built in and, well, just too much stuff to list here. Suffice it to say, you need this add-on. You can get it at our old friend Martin Wright's site (http://fly.to/mwgfx/). Bill's site is down now but Martin has graciously consented to host his files. Follow the links on Martin's page or go directly there using this URL:
http://www.mnwright.btinternet.co.uk/Cfsops/Modules/cfo_weather/Cfo_Weather_75.htm

ActiveSky As described in the sidebar on page 134, ActiveSky is my favorite real time weather simulator. If you can find an older version, such as the 1.4 that I use, it will work perfectly in CFS2. If you can't find it, you might try FSMetar, another real time weather simulator that sounds like it works essentially the same way. FSMetar is Freeware and can be found here: http://personal.telefonica.terra.es/web/fsmetar/

FSSC Flight Simulator Scenery Creator is mandatory if you are serious about creating permanent airbases for entry into CFS2. This is a great freeware program that uses a well designed graphical interface to help you create terrific looking airbases anywhere in the world. It has a bit of a learning curve but is easily mastered with just a little bit of effort. Plus there are help manuals, tutorials, Tips and Tricks, and a user forum at their website. You'll find it at http://www.fssc.avsim.net/.

Any Scenery by Maskrider Maskrider, Chris Westervelt, has produced the best set of Pacific Theater airbases ever made for CFS2. His meticulously detailed and crafted airfields cover a wide range of historical locations scattered throughout the Pacific. Some you've heard of, some are a bit obscure, all are works of art. When Chris' sceneries first starting showing up they injected new life into CFS2 and inspired all of us to try to emulate his mastery of scenery design. His files always include detailed installation instructions and where to find any additional scenery items that he uses, such as Lindsay Watt's trees. If you haven't seen one of his bases yet, you don't know what you're missing. Find his stuff at Sim-Outhouse under the **CFS2 Add-Ons** section. Here's the URL for his website: http://www.mask-rider.com/index.html

SdC World Landclass Scenery Sander de Cocq has done for European scenery what Maskrider did for the Pacific. However, instead of designing new airbases, Sander designed a new Europe! Well, not quite, but he made the stock CFS2 Europe look like the real thing thanks to his new Landclass scenery and terrain textures. Sander's site is no longer available, but fortunately his sceneries are available at Sim-Outhouse. Go to their CFS2 scenery files and search for any of the downloads starting with **SdC**.

Appendix D

Websites worth Visiting

In its heyday, there were more CFS2 websites out there than I could get through in a single day – and that was just the ones I knew about! Over the years, as the sim has aged, many of those sites have fallen by the wayside. Some folded up entirely; some consolidated or were absorbed into other sites; some lie dormant, still there and still containing valuable downloads (thankfully), but they haven't been updated in years. And some sites are still going strong. One in particular is probably better today than it ever was. What's even more encouraging is that in the past year we have seen a couple of new CFS2 sites open up, sites offering fresh and exciting downloads for our favorite flight simulator. Here is a short list of sites I consider worthy of your time. Some of these I visit daily, often again and again to see what's happening out there. Search the Internet and you may find a few gems I overlooked.

One of your best sources of information on the Internet is user forums devoted to CFS2. Forums are great places to learn of the latest developments, get answers to questions and solutions to problems, or just share information and stories about our favorite flight sim. I encourage you to spend time at one or more, especially if you are serious about getting the maximum enjoyment out of CFS2.

Just remember, forums are made up of people just like you and me. All are happy to share their knowledge and most want to be helpful, but their knowledge and experience levels vary widely and there is no way to accurately gauge the quality of the information that you are getting. The length of time a person has been a member of a forum or their number of posts is no guarantee that you are getting the right answer. I don't think anyone would intentionally try to mislead you, but sometimes the answers you get on forums are flat out incorrect. So, use your head, weigh all of your options (you'll find there are many different ways to accomplish the same thing in CFS2), and tread carefully. And always, always, backup before doing anything drastic. If one person suggests you need to do a full reinstallation and someone else recommends just tweaking a file or two, I'd try the tweak first. Always seek the simplest solutions first then work up to the more complex ones.

As you grow more proficient and experienced in CFS2, return the favor and help others who are new to the simulator. Sharing your knowledge and helping others the way you were helped can be very rewarding.

Sim-Outhouse www.sim-outhouse.com I think this is the best place on the Internet to get help on CFS2 and stay up to date on the latest developments. Over the years they have grown into a premier site with excellent forums for all of the Microsoft flight simulators and other aviation related topics. They have a large warehouse of downloads available and the CFS2 forum is populated by some of the most knowledgeable players and designers in the game and Rami does a great job of moderating it. This is one site you want to be sure to Bookmark and visit often.

Simviation www.simviation.com This is one of the granddaddy sites that has been servicing the flight simulator community for years. They have a huge archive of downloads for all of the Microsoft flight sims, from FS98 to FSX, CFS1 to CFS3. This is a great place to go when looking for some older downloads that have dropped off the radar at other sites. They offer both free access and a paid membership. The membership gets you add-free access with faster download speeds, load times and up to date aviation news. They also offer good forums.

Flightsim www.flightsim.com Another one of the granddaddy sites that services all things flight simulator related. They too have a huge archive of downloads plus tons of articles, links and information. They also host a variety of forums. I don't believe they currently offer a CFS2 forum.

Aerocrate www.aerocrate.com If you are even remotely interested in WWI aviation you must bookmark this site. Aerocrate was created by Gary Aumaugher. Gary has turned the CFS2 world clock back to WWI in a way no one else ever has before. There have been WWI airplanes available for download since the beginning, ranging in quality from poor to great, but none with the consistent good looks and feel of Gary's. He is a prolific aircraft designer and we're fortunate he devoted his efforts to this fascinating era in combat aviation. He offers planes, scenery, campaigns and more. Some of the better designers in the business have supported him by offering skins and panels at his website as well. Unfortunately, this is one of those dormant sites that hasn't been updated since 2008. No bother, if you've never been there before everything will seem brand new!

Jaxon's CFS2 Page http://www.michael-reimer.com/CFS2/CFS2.html Michael Reimer (Jaxon) has created an interesting site that is worth visiting. He offers some very good illustrated tips on a variety of CFS2 subjects from simple to complex. His trick of aliasing sounds to an outside folder can save you huge amounts of space on your system (I use the same trick for panels too). He has some neat downloads, such as adding smoke and landing lights to Usio's ships (see below). But, maybe the most interesting thing he offers is a growing collection of WWII aircraft profiles. Axis or Allied, Pacific or Europe, he is pulling together as much data and as many illustrations as he can find. If you're looking to create a texture for a specific unit or pilot, consider looking here first to see if Jaxon has a picture of it.

Usio no Ibuki http://www.geocities.jp/usio_no_ibuki/ This is a Japanese site for CFS2 ships. Usio provides limited English text, but you'll have no trouble figuring out what's what. Bar none, these are the finest, most detailed ships that you will find for CFS2. Most of his ships are Japanese warships but he does offer a few Allied vessels as well. Some ships are offered in multiple versions. For example, you can download both a 1941 and 1944 version of the Japanese battleship Yamato. Beware though, his older models made with FSDS are frame rate killers. If you don't have a fast system, a few of these can turn your game into a slide show. His newer ships are designed in the much more frame rate friendly GMAX and he is converting some of the older ships as well. If you want to flesh out your Japanese fleet, start here. You won't be disappointed.

Hagochu's CFS2 Aircraft Factory http://members.jcom.home.ne.jp/hago/index.htm Another Japanese site, this one specializing in airplanes. What Usio has done for ships, Akemi Mizoguchi has done for airplanes. Although he concentrates on Japanese aircraft, Akemi does offer a few ME-109's and other models as well. His planes are excellently crafted and will help fill the gaps in your hangers. Whether you prefer the Imperial Japanese Army or the Navy, you'll find some great models here, both familiar and less well known. All of his planes have a very realistic look to them as they sit in the sun awaiting your arrival. Like Usio's site, most of the text is in Japanese, but there is enough English thrown in that you won't get lost.

Nanni's CFS Page http://web.tiscalinet.it/Nanni/CFS/ This is an old site that, lucky for us, is still up and running, providing some essential downloadable effects. Nanni, Giovanni Cignoni, only offers about two dozen add-ons here but some, like his All Ships Configuration files are must haves. Nanni and Nibbio have collaborated over the years on several excellent effects which leads us to…

Nibbio's CFS2 Page http://web.tiscalinet.it/jobim/cfs2/ Another Italian based website that, while it hasn't been updated in years, contains some essential CFS2 files that any serious player needs.

Nibbio doesn't offer too many downloads but he has some gems; such as dust clouds for moving vehicles, Japanese and American ground troops that can be shot and killed (they shoot back too!) and a variety of smoke effects. He also offers two essential downloads that Microsoft should have included from the start, the **roll damage fix** and the **Val rear gunner fix** (the latter found on his Tips and Tricks page).

The Ghost Files www.theghostfiles.org Ghost Pony's effects add a new level of realism to CFS2. Although he hasn't added anything new in years, he doesn't need too. Scroll down to the middle of the page and click the link to the CFS2 effects or drop down to the bottom of the page and select one or all of his Combat Paks. You won't be disappointed.

Sopwith C's Paint Shop http://sopwithc.wetpaint.com/ A great site for skins, it combines the best from Sopwith Cameleon and Pombee. Together these two great skinners has amassed one of the best collections of aircraft textures out there. Most are done with multiple variations of a particular aircraft, occasionally with links to where to download the plane. If you want to add new textures, this is a must visit site.

lwdesigns http://forum.lwdesigns.com.au/ This is a new site (2011) that houses a forum and downloads. What makes it special is that it is run by Lindsey Watt. Lindsey is the genius behind the trees and library objects that are essential for most of Maskrider's sceneries mentioned in the must-have downloads section above. In addition he has a variety of new cloud options to try. These are the small, subtle touches that really set your installation off and make CFS2 come alive.

That's just a sampling; there are still dozens more CFS2 sites worth visiting and I'm sure those of you who've been hanging around this sim for years probably have favorites I overlooked. Check out everyone that you find. Occasionally you'll trip across a site with an "essential" download that can't be found anywhere else. Have fun, and watch your six!